AFIT/GE/ENG/99M-14

Optimal Wavelet Denoising for High Range Resolution Radar Classification

THESIS
Brian M. Huether
Second Lieutenant, USAF

AFIT/GE/ENG/99M-14

Approved for public release; distribution unlimited

19990413 100

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 1999	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
Optimal Wavelet Denoising for High Range Resolution Radar Classification

5. FUNDING NUMBERS

6. AUTHOR(S)
Brian M. Huether, Second Lieutenant, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2950 P Street
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/99M-14

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Major Randy Broussard
AFRL/SNZT
2241 Avionics Circle
WPAFB, OH 45433-7318

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Noncooperative identification (NCTI) of airborne targets is a top priority for the Air Force and this thesis makes a significant contribution to that area. High range resolution radar (HRR) provides an attractive means to perform NCTI. When measured HRR data is used, classification performance is excellent. However, it is often not feasible to acquire measured HRR signatures for a wide set of targets, thus necessitating the use of synthetically generated HRR data. Classification performance suffers severe degradation when using synthetic data. This thesis suggests that a large portion of HRR signature content is non-discriminatory and that this content is a cause of classifier degradation for the case of synthetic data. A unique wavelet-based denoising methodology is developed which is optimized with respect to classification accuracy. In the case of synthetic data, the denoising method leads to remarkable classification improvements. Classification accuracies are obtained which are comparable to those when training on measured data. This is an unprecedented result. It is also shown that the denoising approach of this thesis leads to superior classification results compared to those obtained with traditional wavelet-based methods.

14. SUBJECT TERMS
Wavelets, Denoising, High Range Resolution Radar (HRR), Pattern Recognition, Classification

15. NUMBER OF PAGES
135

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
unclassified	unclassified	unclassified	UL

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

AFIT/GE/ENG/99M-14

Optimal Wavelet Denoising for High Range Resolution Radar Classification

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Brian M. Huether, B.S. Electrical Engineering
Second Lieutenant, USAF

March 1999

Approved for public release; distribution unlimited

AFIT/GE/ENG/99M-14

Optimal Wavelet Denoising for High Range Resolution Radar Classification

Brian M. Huether, B.S. Electrical Engineering

Second Lieutenant, USAF

Approved:

	5 Mar 99
Dr. Steven Gustafson Thesis Advisor	Date
	5 MAR 99
Dr. Steven Rogers Committee Member	Date
Mark Oxley	5 Mar 99
Dr. Mark Oxley Committee Member	Date
Randy Broussard	5 MAR 99
Maj. Randy Broussard Committee Member	Date

Acknowledgements

There are a number of people who deserve recognition in regards to my thesis. First, I would like to thank my thesis advisor Dr. Steve Gustafson for allowing me to execute this thesis with such a large degree of freedom. I am sure there must have been moments when Dr. Gustafson thought to himself, "I wonder just what it is that Brian is doing … " Believe me - there were moments when I had the same thought. My committee members also deserve thanks: Dr Steve Rogers, who introduced me to the challenging field of pattern recognition when I was an intern at AFIT, made me realize that there are more than 24 hours in a day if you really need there to be! Major Randy Broussard played a duel role of sponsor and committee member and helped me establish the challenging goals that this thesis satisfied. Dr. Mark Oxley deserves thanks for demonstrating patience with me as I would come practically barging into his office rambling incoherently about the peculiarities of wavelet analysis. Also deserving thanks are Pete Kosir and Rob Dewall. They were not committee members, but they may as well have been, for they provided me with a great deal of technical guidance and helped keep me on track. I want to thank all users of the Hawkeye Computer Lab for not ganging up on me, as I am sure each and every one of them must have noticed my extraordinarily excessive usage of the Hawkeye computers. Dave Doak also deserves thanks in this context, for he provided invaluable UNIX support and also allowed me to use a certain, "special computer." Along the same lines I would like to thank Major Scott Berger for allowing me to take advantage of the meanest machine of them all. Some of my last minute, computationally intensive thesis work was made possible by the use of his mean machine. Dr. William Brown deserves special thanks. Though he was not involved in my thesis work, he inspired me in ways for which I will be forever grateful. He made me realize that mathematics is just as much an art and a philosophy, as it is a science. I only hope that someday I am capable of the mental gymnastics that Dr. Brown performs with such ease. I need to thank my partners in crime, Dan Petrovich and Barry Vanek. We all shared the same philosophy while at AFIT - You have to have fun in life - and we made sure that philosophy was upheld as often as possible. My parents, brothers, and relatives also deserve thanks because they always welcomed me home during those between quarter breaks. Lastly, and most importantly, I

want to thank my one-in-a-billion girlfriend Kathy for being so patient and understanding during those times when it was a struggle for me to simply to remember my own name! I also want to thank her for pointing out to me the subtleties of classical music which helped me cast intuition upon pattern recognition theory. There are an unlimited number of other reasons I have to thank her, but they transcend the expressiveness of mere words.

Brian M. Huether

Table of Contents

Page

Acknowledgements iii

List of Figures viii

List of Tables xiii

Abstract xvii

I. Introduction 1-1
- 1.1 Background 1-1
- 1.2 Review of HRR Classification Schemes 1-3
- 1.3 Problem Statement 1-4
- 1.4 Scope 1-6
- 1.5 Methodology 1-7
- 1.6 Objectives 1-8
- 1.7 Organization 1-8

II. Theory 2-1
- 2.1 Introduction 2-1
- 2.2 Pattern Recognition 2-1
  - 2.2.1 Bayesian Decision Theory 2-1
  - 2.2.2 Discriminant Functions 2-3
  - 2.2.3 Feature Extraction and Pre-Processing 2-7
- 2.3 Wavelet Analysis 2-8
  - 2.3.1 Multiresolution Analysis 2-11
  - 2.3.2 Daubechies Wavelets 2-18
  - 2.3.3 Wavelet Denoising 2-20
- 2.4 Summary 2-30

Page

III. Methodology 3-1

3.1 Introduction 3-1

3.2 Baseline Gaussian Classifier 3-2

3.2.1 Classifier Training 3-3

3.2.2 Classifier Testing 3-4

3.3 Wavelet Denoising 3-4

3.3.1 Wavelet Selection 3-8

3.3.2 Choice of Coarsest Scale 3-9

3.3.3 Hard or Soft Thresholding 3-9

3.3.4 Applying Thresholds to TI Table 3-10

3.3.5 Threshold Selection 3-13

3.3.6 Optimization 3-13

3.3.7 Procedure 3-17

3.4 Summary 3-18

IV. Results 4-1

4.1 Introduction 4-1

4.2 Single Window 4-1

4.2.1 Training on Measured Data 4-1

4.2.2 Training on Synthetic Data 4-12

4.3 Multiple Windows 4-20

4.3.1 Training on Measured Data 4-21

4.3.2 Training on Synthetic Data 4-27

4.4 Additional Considerations 4-34

4.4.1 Performance with Larger testing Sets 4-35

4.4.2 Wavelet Sensitivity 4-38

4.4.3 Alternative Denoising Methods 4-40

4.4.4 Variation of the Denoising Implementation 4-44

Page

4.4.5 Alternative Optimization Procedure for Multiple Windows 4-46

4.5 Summary 4-49

V. Conclusions and Recommendations 5-1

5.1 Introduction 5-1

5.2 Summary of Key Results 5-1

5.3 Recommendations for Future Work 5-2

5.4 Summary 5-4

Bibliography BIB-1

Vita VITA-1

List of Figures

Figure Page

1.1. Measured HRR signature (top) and synthetic signature (bottom) both from same target and 5 X 5 degree (azimumth X elevation) window. Magnitudes are normalized. 1-2

1.2. Illustration of the HRR radar data distribution. 1-6

1.3. Visualization of measured HRR signatures collected from a 5 X 5 window. 1-7

1.4. Visualization of synthetic HRR signatures collected from a 5 X 5 window. 1-8

2.1. Illustration of the joint probability densities for two classes as a function of a feature x. The vertical line indicates a sub-optimal decision boundary and the arrow indicates an optimal decision boundary. . . 2-4

2.2. Resolution cells in time-frequency plane of short time Fourier transform. 2-9

2.3. Resolution cells in time-frequency plane of wavelet transform. . . . 2-10

2.4. Nested spaces spanned by scaling functions. 2-13

2.5. Nested spaces spanned by scaling functions and wavelets. 2-13

2.6. Signal projected onto the scaling function spaces. 2-16

2.7. Signal projected onto the wavelet function spaces. 2-17

2.8. Filter bank implementation of discrete time wavelet decomposition. 2-18

2.9. Filter bank implementation of discrete time wavelet reconstruction. 2-19

2.10. Wavelet and scaling functions for $daub_4, daub_{10}$, and $daub_{16}$. 2-21

2.11. The Haar wavelet and scaling function. 2-22

2.12. Illustration of soft and hard thresholding. 2-23

2.13. Filter block for TI wavelet transform implementation. 2-27

3.1. Block diagram of baseline classifier with denoising process indicated. 3-2

Figure Page

3.2. Unprocessed HRR signature (left) and pre-processed HRR signature (right). 3-4

3.3. Templates for all targets after pre-processing. 3-5

3.4. Illustration of the testing process of the baseline classifier. 3-6

3.5. Fully translation invariant projection of measured HRR signature onto scaling function spaces. 3-10

3.6. Fully translation invariant projection of synthetic HRR signature onto scaling function spaces. 3-11

4.1. Baseline classification accuracies versus number of looks for the case of a single window and measured training data. 4-3

4.2. Illustration of target separability and inseparability: (a) Complete separability of targets A and E; (b) Inseparability leading to misclassifications of target D 4-4

4.3. Visualization of maximum classification accuracies as a function of threshold pairs for the case of a single window and measured training data. (a) Decomposition level 1; (b) Decomposition level 2; (c) Decomposition level 3 4-6

4.4. (a) HRR signature; (b) Optimally Denoised signature; (c) Sub-optimally denoised signature 4-8

4.5. Denoised signal representations for the case of a single window and measured training data: (a) Original HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signatures using level 2 parameters; (d) Denoised signature using level 3 parameters . . . 4-9

4.6. Target accuracies with denoising versus the number of looks for the case of a single window and measured training data. (a) Level one denoising performance (b) Level three denoising performance 4-11

4.7. Baseline target accuracies versus number of looks for the case of a single window and synthetic training data. 4-13

4.8. Visualization of overall target accuracies as a function of threshold pairs for the case of a single window and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2; (c) Decomposition level 3 4-15

Figure Page

4.9. Denoised signal representations for the case of a single window and synthetic training data: (a) Original synthetic HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signatures using level 2 parameters; (d) Denoised signature using level 3 parameters 4-16

4.10. Denoising performance versus the number of looks for the case of a single window and synthetic training data. 4-17

4.11. Scatter plots before and after denoising for the case of a single window, ten looks, and synthetic training data. 4-18

4.12. On the left, are original templates along with a test signature. On the right are templates formed by denoising along with a denoised test signature. 4-19

4.13. Scatter plots before and after denoising for the case of a single window, ten looks, and synthetic training data. 4-20

4.14. Average baseline target accuracies versus number of looks for the case of five windows and measured training data. 4-23

4.15. Visualization of average overall target accuracies as a function of threshold pairs for the case of multiple windows and measured training data. (a) Decomposition level 1; (b) Decomposition level 2; (c) Decomposition level 3 4-25

4.16. Denoised signal representations for the case of five windows and measured training data: (a) Original measured HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signatures using level 2 parameters; (d) Denoised signature using level 3 parameters 4-26

4.17. Average target accuracies with denoising versus number of looks for the case of five windows and measured training data. 4-27

4.18. Average baseline target accuracies versus number of looks for the case of five windows and synthetic training data. 4-31

4.19. Visualization of average overall target accuracies as a function of threshold pairs for the case of multiple windows and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2; (c) Decomposition level 3 4-32

Figure Page

4.20. Denoised signal representations for the case of five windows and synthetic training data: (a) Original measured HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signatures using level 2 parameters; (d) Denoised signature using level 3 parameters 4-33

4.21. Average target accuracies with denoising versus number of looks for the case of five windows and synthetic training data. 4-34

4.22. Span of HRR data used for testing. 4-39

4.23. Illustration of the sensitivity of classification accuracy with respect to the wavelet choice for the case of a single window, single look, and synthetic training data. 4-41

4.24. Maximum VisuShrink and SureShrink overall target accuracy as a function of decomposition level for the case of a single window and synthetic training data. Top line corresponds to maximum accuracy achieved through TI denoising; bottom line corresponds to baseline accuracy. (a) Single look; (b) Ten looks 4-42

4.25. Denoised signal representations obtained through VisuShrink for the case of a single window and synthetic training data. (a) Original HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signature using level 4 parameters; (d) Denoised signature using level 7 parameters; 4-43

4.26. Denoised signal representations obtained through SureShrink for the case of a single window and synthetic training data. (a) Original HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signature using level 4 parameters; (d) Denoised signature using level 7 parameters; 4-44

4.27. Visualization of overall classification accuracies obtained with FIR filtering for the case of a single window, single look, and synthetic training data. 4-45

4.28. Original HRR signature (top) and denoised signature (bottom) obtained with optimal FIR filtering. 4-45

4.29. Level one denoising accuracy surfaces for the case of a single window and synthetic training data. Left: TI method; Right: Non-TI method 4-46

Figure Page

4.30. Visualization of maximum classification accuracies as a function of threshold pairs for the case of $win_{60,15}$ and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2 4-47

4.31. Visualization of maximum classification accuracies as a function of threshold pairs for the case of $win_{60,25}$ and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2 4-47

4.32. Visualization of maximum classification accuracies as a function of threshold pairs for the case of $win_{70,15}$ and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2 4-48

4.33. Visualization of maximum classification accuracies as a function of threshold pairs for the case of $win_{75,15}$ and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2 4-48

List of Tables

Table Page

1.1. Summary of HRR Classification Approaches 1-4

1.2. Summary of HRR Classification Approaches (continued) 1-5

3.1. Target subsets of interest 3-15

3.2. Single window cases using $win_{65,15}$ 3-16

3.3. Multiple window cases using $win_{60,15}$, $win_{60,25}$, $win_{65,15}$, $win_{70,15}$, and $win_{75,15}$ 3-17

4.1. Baseline target confusion matrix for the case of a single window, single look, and measured training data 4-2

4.2. Baseline target confusion matrix for the case of a single window, ten looks, and measured training data. 4-5

4.3. Optimal denoising parameters for the case of a single window and measured training data 4-9

4.4. Target accuracies with denoising for the case of a single window, single look, and measured training data. 4-10

4.5. Target accuracies with denoising for the case of a single window, ten looks, and measured training data. 4-10

4.6. Target confusion matrix with denoising for the case of a sinle window, ten looks, and measured training data. 4-11

4.7. Relative classification improvements for the case of a single window and measured training data. 4-12

4.8. Baseline target confusion matrix for the case of a single window, single look, and synthetic training data. 4-12

4.9. Baseline target confusion matrix for the case of a single window, ten looks, and synthetic training data. 4-14

4.10. Optimal denoising parameters for the case of a single window and synthetic training data. 4-14

Table Page

4.11. Target accuracies with denoising for the case of a single window, single look, and synthetic training data. 4-15

4.12. Target accuracies with denoising for the case of a single window, ten looks, and synthetic training data. 4-16

4.13. Target confusion matrix with denoising for the case of a single window, ten looks, and synthetic training data. 4-17

4.14. Relative classification improvements for the case of a single window and synthetic training data. 4-21

4.15. Baseline target accuracies for the case of five windows, a single look, and measured training data. 4-22

4.16. Cummulative baseline target confusion matrix for the case of five windows, a single look, and measured training data. 4-22

4.17. Baseline target accuracies for the case of five windows, ten looks, and measured training data. 4-23

4.18. Cummulative baseline target confusion matrix for the case of five windows, ten looks, and measured training data. 4-24

4.19. Optimal denoising parameters for the case of five windows and measured training data. 4-24

4.20. Average target accuracies with denoising for the case of five windows, a single look, and measured training data. 4-25

4.21. Average target accuracies with denoising for the case of five windows, ten looks, and measured training data 4-26

4.22. Cummulative target confusion matrix with denoising for the case of five windows, ten looks, and measured training data. 4-28

4.23. Target accuracies with denoising for the case of five windows, ten looks, and measured training data 4-28

4.24. Relative classification improvements for the case of five windows and measured training data. 4-29

4.25. Baseline target accuracies for the case of five windows, a single look, and synthetic training data. 4-29

Table Page

4.26. Cummulative baseline target confusion matrix for the case of five windows, a single look, and synthetic training data. 4-30

4.27. Baseline target accuracies for the case of five windows, ten looks, and synthetic training data 4-30

4.28. Cummulative baseline target confusion matrix for the case of five windows, ten looks, and synthetic training data. 4-31

4.29. Optimal denoising parameters for the case of five windows and synthetic training data. 4-32

4.30. Average target accuracies with denoising for the case of five windows, a single look, and synthetic training data. 4-33

4.31. Average target accuracies with denoising for the case of five windows, ten looks, and synthetic training data 4-34

4.32. Target accuracies with denoising for the case of five windows, ten looks, and synthetic training data 4-35

4.33. Cummulative target confusion matrix with denoising for the case of five windows, ten looks, and synthetic training data. 4-35

4.34. Relative classification improvements for the case of five windows and synthetic training data. 4-36

4.35. Baseline target confusion matrix for the case of a single window, single look, synthetic training data, and all available testing data. 4-36

4.36. Baseline target confusion matrix for the case of a single window, ten looks, synthetic training data, and all available testing data. 4-37

4.37. Target confusion matrix with denoising for the case of a single window, single look, synthetic training data, and all available testing data. . 4-37

4.38. Target confusion matrix with denoising for the case of a single window, ten looks, synthetic training data, and all available testing data. . . 4-38

4.39. Relative classification improvements for the case of five windows, synthetic training data, and all available testing data. 4-38

4.40. Cummulative baseline target confusion matrix for the case of 12 windows, one look, and synthetic training data. 4-38

Table Page

4.41. Cummulative baseline target confusion matrix for the case of 12 windows, ten looks, and synthetic training data. 4-39

4.42. Cummulative target confusion matrix with denoising for the case of 12 windows, one look, and synthetic training data. 4-39

4.43. Cummulative target confusion matrix with denoising for the case of 12 windows, ten looks, and synthetic training data. 4-40

4.44. Relative classification improvements for the case of 12 windows, synthetic training data, and all available testing data. 4-40

4.45. Optimal level two denoising parameters for the case of five windows and synthetic training data 4-49

4.46. Comparison of overall target accuracies for different multiple window optimization techniques 4-49

AFIT/GE/ENG/99M-14

Abstract

Noncooperative target identification (NCTI) is a top priority for the Air Force, with emphasis on airborne targets. Key achievements of this thesis include unprecedented results that make a significant contribution towards the aims of NCTI.

High range resolution radar has received a significant amount of attention due to its ability to resolve closely spaced scatterers on a target. Processing the backscattered radar energy yields a target signature which then forms the basis for template-based classification. When measured signatures are used for classifier training, classification performance is excellent. However, it is often unfeasible to acquire measured HRR signatures for a wide set of targets, thus necessitating the use of synthetically generated HRR data. This data is used to create target templates for comparison with measured signatures. Classification performance suffers severe degradation when using the synthetic data for template formation. A goal of any HRR classification system then is to improve classification accuracy when using synthetic data, ultimately enabling equivalent performance to that of measured data.

This thesis suggests that a large portion of HRR signature content is non-discriminatory and that this content is a cause for classifier degradation for synthetic training data. We view this content as a form of abstract noise, and thus treat the classification problem in the context of noise removal. Well-established wavelet denoising methods have proved to be superior for signal denoising. However, these powerful methods assume a Gaussian noise model and are optimized with respect to a risk measure which often takes the form of mean squared error. The abstract notion of noise makes these wavelet methods unsuitable, and so a unique wavelet-based denoising methodology is developed which is optimized with respect to classification accuracy.

In the case of synthetic training data, the denoising method of this thesis leads to remarkable classification improvement. In particular, we obtain classification accuracies which are comparable to those obtained when training on measured data. This is an unprecedented result. We also show that the denoising approach of this thesis leads to superior results compared to those obtained with standard wavelet-based approaches.

Optimal Wavelet Denoising for High Range Resolution Radar Classification

I. Introduction

1.1 Background

Combat identification (CID) plays a major role in today's military and new technology in this area is being actively pursued by the Air Force. Reliable aircraft identification has typically required the use of Identify Friend or Foe (IFF) systems. For such a system to work, the aircraft to be identified must to be equipped with the IFF system. Clearly this limits the scope of aircraft identification, because it assumes cooperation between the identifier and the aircraft being identified. A more general approach is Noncooperative Target Identification (NCTI). A NCTI system performs identification essentially by means of remote sensing with no coordination with the target being sensed. Top level priorities for such a system are the destruction of hostile targets and the preservation of non-combatant, neutral, and friendly targets. The system must provide a declaration of target type and do so with confidence such that the previously mentioned priorities can be upheld.

The primary sensor for CID is radar with a concentration on tactical airborne radar systems which enable the active or passive collection of multimode electromagnetic data. The CID function is then typically carried out by template matching (16). A promising sensor is High Range Resolution Radar (HRR). Range resolution is the ability of the radar to resolve point targets that are separated in range to the radar. In general, HRR works by illuminating a target with wideband radar energy and processing the backscattered energy. The range resolution is $\Delta r_s \approx c/2\beta$, where c is the speed of light and β is the radar bandwidth. It is the large β characteristic of HRR radar that enables high range resolution. After processing, a target signature is formed which measures energy as a function of range. Thus, individual scatterers of a target contribute to the signature in such a way that a signature is characteristic of a target. Although it was thought that most major backscatter sources at any given target aspect are produced by specular reflection from flat surfaces of the target normal to the radar and from corners, it is now known that other mechanisms contribute. Radar energy actually propagates along the surface of some part of the target and reappears directed toward the radar. This phenomena is known as *creeping wave* reflection. Resonance effects can also produce reflections (46). Figure 1.1 shows a typical HRR signature for a target. This signature consists of energies across 461 range bins, each of which can be considered a feature. The peaks in the signal represent prominent features of the aircraft.

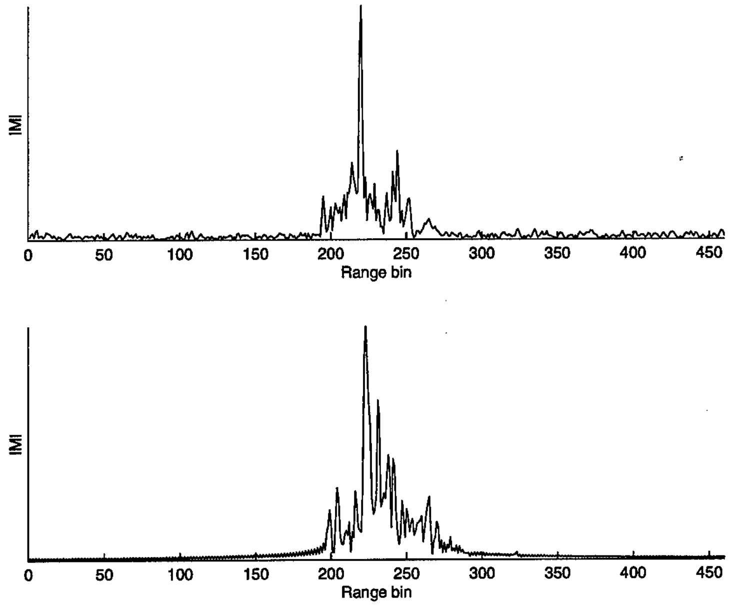


Figure 1.1 Measured HRR signature (top) and synthetic signature (bottom) both from same target and 5 X 5 degree (azimumth X elevation) window. Magnitudes are normalized.

The ability of HRR to resolve closely spaced features can, ironically, be problematic. The high range resolution causes significant changes in a target's signature when the target's orientation changes. Thus, small azimuth and elevation changes lead to considerable variability in the signatures for a given aircraft. This variability can be attributed to the coherent interactions of the backscattered radar energy from many scatterers. When a target is moving, energy reflected from these scatterers moves in and out of phase, causing constructive and destructive interference that results in fluctuations in the total return amplitude. The implications of this variability are discussed thoroughly in (13). There is a trade-off that must be accepted, and as a consequence HRR classifiers need to have added functionality to account for the high degree of signature variability. The added functionality amounts to sectoring the data into windows in which there is a small azimuth and elevation change and constructing templates for all sectors.

For CID systems to operate as described above, databases must exist that contain a sufficient number of radar signatures collected for the various targets that the system is to identify. It is not always feasible to obtain signatures for a target; for example, foreign nations may deny the United States the ability to collect radar signatures for targets. For this reason, there is a strong motivation within the Department of Defense to generate radar signatures synthetically. Through intelligence and other means, accurate models of targets can be constructed, and electromagnetic simulation software can then synthesize the radar signatures. These synthetic signatures can then be used in the same way as their measured counterparts - they form templates for a matching process. The ability to generate synthetic signatures also has the added benefit of reduced expense. However, this paradigm is inherently problematic since the synthetic signatures do not look precisely like the real ones due to the inaccuracies introduced by the modeling. Thus it is not surprising that classification performance is degraded when forming synthetic templates. Further research and development is necessary to successfully incorporate synthetic models into the overall CID philosophy.

1.2 Review of HRR Classification Schemes

As with any classification problem, HRR classification approaches vary widely. Tables 1.1 and 1.2 summarize some of the approaches taken over the past decade. (This listing is by no means comprehensive.) From the tables, we see that the HRR classification approaches have covered a wide spectrum, ranging from very simple to very complex. Note also that feature extraction often amounts to simply retaining the HRR signatures. Since no reliable feature sets have been found for HRR classification, it is sensible to use the raw data. The inability to find reliable features can be attributed to the variability of HRR signatures as mentioned above.

Reference	Feature Extraction	Classification
(43),1992	HRR signatures	Gaussian and Synthetic Discriminant Classifier
(55),1994	HRR signatures	Maximum Liklihood Bayes Classifier
(24), 1994	HRR range bin selection using decision boundary analysis	K-Nearest Neighbor, Multilayer Perceptron, Gaussian Classifier
(3), 1995	HRR signatures represented in scale space using wavelet decomposition	Tree-Structured Vector Quantization
(41), 1995	Obtain relative range, size and shape of scattering centers	Modified Correlation Method
(45), 1995	Retain middle 64 range bins of HRR signatures	Radial Basis Function Classifier
(49), 1996	Logarithms of energies of HRR decomposition at adjacent wavelet scales	Radial Basis Function Classifier
(52), 1996	HRR signatures	Modified LVQ2
(28), 1996	HRR signatures	Adaptive Matched Filter
(17), 1996	HRR signatures	Multilayer Perceptron

Table 1.1 Summary of HRR Classification Approaches

1.3 Problem Statement

This thesis investigates the use of wavelet-based denoising as a means to improve HRR classification accuracy. The denoising acts merely as a pre-processing step which is used in conjunction with a Gaussian classifier. In particular, we are interested in the case of training on synthetic data, for it is with this case that current performance is severely lacking. When training on measured data, current performance is excellent and we therefore do not expect to achieve considerable improvement. However, we certainly want the performance when training on measured data to at least match current performance.

The desire to perform denoising stems from an intuition gathered by visual examination of the raw HRR signatures. The jagged appearance of the HRR signatures is reminiscent of high order polynomial fitting to measured data samples, in which case the polynomial overfits to the noise of the underlying signal. In the spirit of Occam's razor, which states we should prefer simple models to complex ones (4), we then desire to trans-

Reference	**Feature Extraction**	**Classification**
(18), 1996	Fractal dimension of HRR signatures	Fractal dimensions used to discriminate amongst targets
(53), 1996	Wavelet detail representation of HRR signatures at various scales	Nearest neighbor
(30), 1997	Thresholded HRR signatures	Time Delay Neural Network
(25), 1997	Spectral peaks extracted from HRR ARMA models at various scales	Minimum Distance Classifier
(51), 1997	Eight HRR signature peaks with highest energy kept using geometry information	Radial Basis Function Classifier
(29), 1997	Target length, maximum amplitude, symmetry, and moments extracted from HRR signatures	Matched Filter, NNC, MLP, RBF
(50), 1997	Transient polarization response	Multiresolution Neural Network
(33), 1997	Low frequency Fourier coefficients of HRR signatures	Hidden Markov and Gaussian Mixture Models
(40), 1997	HRR signatures	Vector Quantizer, Kohonen Feature Map, Gaussian Classifier
(54), 1998	HRR signatures	Gaussian and Multinomial Pattern Matching Classifier

Table 1.2 Summary of HRR Classification Approaches (continued)

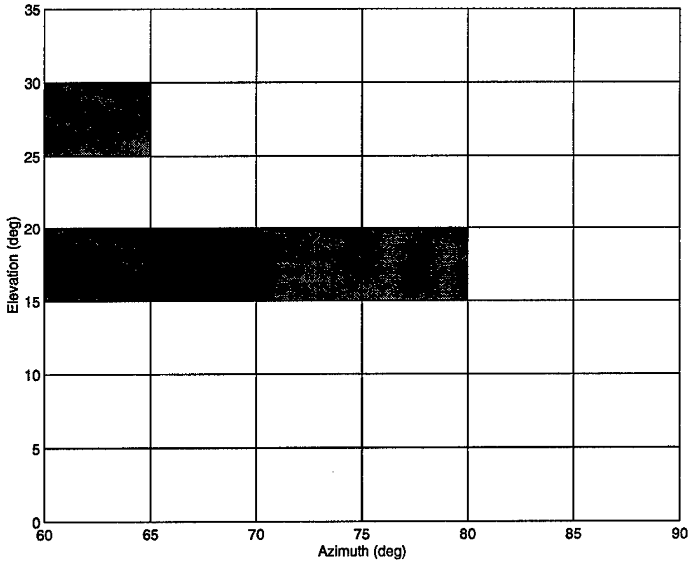


Figure 1.2 Illustration of the HRR radar data distribution.

form the raw HRR signatures such that their representation becomes simpler. We are also interested in evaluating the generalization properties of the denoising/classification scheme, using an appropriate measure of "generalization".

1.4 Scope

There are two databases available for use in this thesis. One is a six class database of measured HRR signals which span 60-90° in azimuth and 0-35° in elevation. The other database covers the same targets and the same span but contains synthetic data. Among these six targets are three "easy" and three "hard" targets. Our interest lies with the full set of targets and all of our effort will be directed towards the full target set.

Figure 1.2 illustrates the distribution of the HRR data; the shaded regions indicate data rich sectors. We see that there are 42 windows each 5 X 5 degrees in size. Figure 1.3 shows 25 signatures for a particular aircraft, taken from one 5 X 5 window. The signatures are reordered from the most similar to the most dissimilar from the mean of the signatures, and only range bins 150-300 are shown, which has been done simply for visualization purposes. Figure 1.4 shows all 25 synthetic signatures for the same aircraft and same 5 X 5 window; reordering has been done in the same manner as for the measured

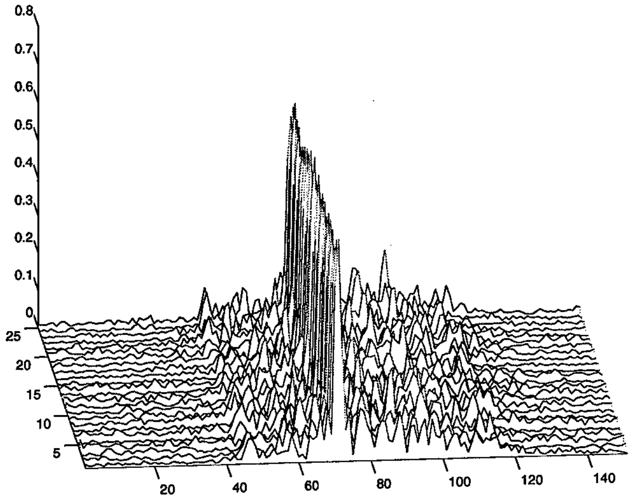


Figure 1.3 Visualization of measured HRR signatures collected from a 5 X 5 window.

signatures. The variability across the measured and synthetic signatures gives us an idea of how potentially difficult HRR classification can be, though an underlying structure can be discerned (e.g., the dominant peak in the middle).

Ideally, each window would be data rich, but due to the manner in which data is collected, certain windows contain several hundred signatures, while others contain several dozen. Furthermore, for some targets there are small numbers of signatures across all windows. We want to perform classification with a sufficient number of signatures in a given window for *all* classes, so we choose one window meeting this criteria, and then add several others to evaluate the generalization of the classification method.

1.5 Methodology

The approach is two fold. First we develop a wavelet-based denoising scheme which is incorporated into the baseline classifier. This denoising scheme is then optimized such that for a given set of classification parameters, maximum classification accuracy is obtained. The optimization process determines the denoising parameters. Using this approach, we optimize for the case of a single 5 X 5 window and then extend the method to multiple 5 X 5 windows. In each case we are interested in training on real and synthetic data, and on comparing the results with those of the current classifier.

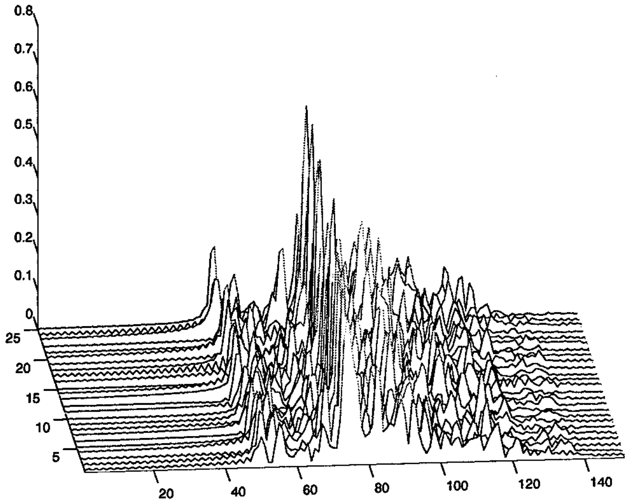


Figure 1.4 Visualization of synthetic HRR signatures collected from a 5 X 5 window.

1.6 Objectives

There are 6 objectives:

1. Provide the reader with a sufficient coverage of relevant theory.
2. Develop a thorough wavelet denoising methodology.
3. Demonstrate that for the case of training on measured data, proper denoising can lead to equivalent classification accuracy when applied to a single window.
4. Demonstrate that for the case of training on synthetic data, proper denoising can lead to increased classification accuracy when applied to a single window.
5. Extend classification to multiple windows to suggest generalization of results beyond the single window case.
6. Demonstrate that the developed denoising method leads to superior results compared to the methods prevalent in the wavelet literature.

1.7 Organization

Chapter 2 covers the necessary theoretical material, which consists of statistical pattern recognition and wavelet analysis. Chapter 3 sets the foundation for our wavelet denoising scheme. Comprehensive results are presented in chapter 4. Chapter 5 provides conclusions and recommendations for future work.

II. Theory

2.1 Introduction

In this chapter we cover the theory relevant for this thesis. We introduce statistical pattern recognition from a Bayesian viewpoint and then cover wavelet analysis and powerful wavelet-based denoising techniques. The theory is presented such that it is independent from the subject matter of the thesis and thus it will serve in a stand-alone fashion.

2.2 Pattern Recognition

Consider for a moment, qualitatively, the manner in which humans recognize patterns and objects. If you are a classical music aficionado, then you can distinguish a composition from the Baroque era and one from the Romantic era, because you have listened to enough examples from both eras and have a "feel" for the unique sounds of both. The problem of determining from which era a certain piece of classical music is can be stated in a more formal way as follows: Given the characteristics of a classical music piece, from what era is it? Of course, it is unreasonable to assert that this decision could be made with 100 % certainty, and so the problem must be treated from a probabilistic standpoint, and may now be stated as: Given the characteristics of the classical music piece, what era was it *most likely* from? We can make a slight modification to this line of thinking which enables us to obtain a firmer grasp of "most likely." Suppose that the number of musical pieces composed during the Romantic era exceeded the number composed during the Baroque era. This type of knowledge can be of great utility. If a decision had to be made before listening to the composition (an unfair circumstance!), then the wise thing to do would be to decide that the piece was from the Romantic era. Once we listen to the piece, then to make a decision we can use the characteristics of the music *and* knowledge regarding the relative numbers of compositions from each era. This qualitative analysis leads to the Bayesian formalism of pattern recognition discussed below.

2.2.1 Bayesian Decision Theory. We now consider a general classification problem in which we try to classify N objects. Let $P(C_k)$ be the fraction of all objects in the k^{th} object class. These fractions are the *a priori* probabilities. Let $p(\mathbf{x}|C_k)$ be the *class* *conditional density* of the D-dimensional feature vector $\mathbf{x}$ for class k. Finally, define $p(\mathbf{x})$ as the unconditional density of $\mathbf{x}$. Bayes Theorem can then be expressed as

$$P(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)P(C_k)}{p(\mathbf{x})}, \tag{2.1}$$

where $P(C_k|\mathbf{x})$ is the *posterior probability* for class k. The unconditional density is written as

$$p(\mathbf{x}) = \sum_{k=1}^{N} p(\mathbf{x}|C_k)P(C_k), \tag{2.2}$$

and so it acts as a normalization factor and can be disregarded for the decision making process. The left side of Equation (2.1) can be read as "Given the feature vector $\mathbf{x}$, what is the probability that it belongs to class k?" Intuitively, we arrive at the following decision rule: A feature vector $\mathbf{x}$ is labeled as class k if

$$P(C_k|\mathbf{x}) > P(C_j|\mathbf{x}) \text{ for all } k \neq j. \tag{2.3}$$

By noting the unimportance of $p(\mathbf{x})$ and by making use of Equation (2.1), we can rewrite Equation (2.3) as

$$p(\mathbf{x}|C_k)P(C_k) > p(\mathbf{x}|C_j)P(C_j) \text{ for all } k \neq j. \tag{2.4}$$

Since the classifier assigns each feature vector to one of N classes, we envision the feature space as being composed of *decision regions* $\mathcal{R}_1, ..., \mathcal{R}_N$. We must determine the placement of the *decision boundaries* such that the probability of misclassification is minimized. Consider a problem with two classes and one dimensional feature vectors. A misclassification occurs when a feature vector is assigned to class C_1 when the true class is C_2 and vice versa. We can write the total probability of error as

$$\begin{aligned} P(error) &= P(\mathbf{x} \in \mathcal{R}_2, C_1) + P(\mathbf{x} \in \mathcal{R}_1, C_2) \\ &= P(\mathbf{x} \in \mathcal{R}_2 | C_1) P(C_1) + P(\mathbf{x} \in \mathcal{R}_1 | C_2) P(C_2) \\ &= \int_{\mathcal{R}_2} p(\mathbf{x}|C_1) P(C_1) + \int_{\mathcal{R}_1} p(\mathbf{x}|C_2) P(C_2), \end{aligned} \tag{2.5}$$

where $P(\mathbf{x} \in \mathcal{R}_1, C_2)$ is the joint probability of assigning a feature vector to class C_1 and having a true class of W_2. If $p(\mathbf{x}|C_1)P(C_1) > p(\mathbf{x}|C_2)P(C_2)$ for a given $\mathbf{x}$, then $\mathcal{R}_1$ and $\mathcal{R}_2$ should be chosen so that $\mathbf{x}$ lies in $\mathcal{R}_1$, since this choice will yield a smaller contribution to the error and is precisely the decision rule given by Equation (2.4). If the joint probability densities are as in Figure 2.1 and the decision boundary is located at the vertical line, then we would have a sub-optimal classifier, since the shaded regions correspond to misclassifications. If instead we place the decision boundary at the point where the densities cross as indicated by the arrow, then we would minimize the area of the shaded region, thus minimizing the probability of misclassification. Equation (2.4) would have us do so and this placement is equivalent to assigning a feature vector to the class for which it has the largest posterior probability. This decision rule readily extends to the general case of N classes and D-dimensional feature vectors.

2.2.2 Discriminant Functions. We saw in the previous section that class assignment is based on the relative sizes of the probabilities. This fact allows us to use a set of *discriminant functions* to perform classification. Each class has a discriminant function $y_k(\mathbf{x})$ such that the following decision rule can be used: Assign a feature vector $\mathbf{x}$ to class C_k if

$$y_k(\mathbf{x}) > y_j(\mathbf{x}) \quad \text{for all } k \neq j. \tag{2.6}$$

Equation (2.4) shows how to choose the discriminant functions: We choose them so that

$$y_k(\mathbf{x}) = p(\mathbf{x}|C_k) P(C_k). \tag{2.7}$$

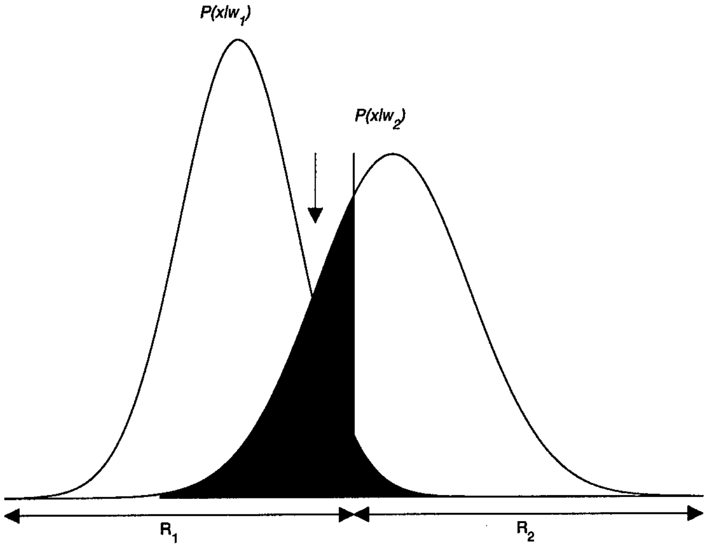


Figure 2.1 Illustration of the joint probability densities for two classes as a function of a feature x. The vertical line indicates a sub-optimal decision boundary and the arrow indicates an optimal decision boundary.

The relative magnitudes of the discriminant functions determine class assignments, so we can use a monotonic function to perform the transformation $g(\,y_k(\mathbf{x}))$. This transformation has the important property of preserving decision region boundaries. A particularly useful monotonic function is the natural logarithm function. Applying the natural logarithm to Equation (2.7) yields

$$y_k(\mathbf{x}) = \ln p(\mathbf{x}|C_k) + \ln P(C_k). \tag{2.8}$$

We have seen from the preceding sections that probability densities play a vital role in Bayesian decision theory. Some controversy arises over the Bayesian approach because it assumes that the densities are available; whether or not the densities you are using are representative of the underlying densities is the issue. Thus, density estimation is an important preliminary step before classification. There are two general methods used to estimate densities. The parametric method assumes a functional form for the density and estimates the parameters that define the functional form. An alternative approach is the non-parametric method, which assumes no functional form but fits a density to the data. A histogram is a crude non-parametric method. Though not used in practice, it forms the basis for more justifiable non-parametric density estimation methods ((4) provides a sufficient overview of both density estimation techniques as applied to pattern recognition).

2.2.2.1 Quadratic Classifier. We now examine a particular classifier based on the preceding ideas. First we define the Gaussian density. For a one dimensional feature vector x, the Gaussian density function is

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \tag{2.9}$$

where μ and σ^2 are the mean and variance of the feature, respectively. The Gaussian density extends to feature vectors of D dimensions and has the form

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\}, \tag{2.10}$$

where $|\mathbf{A}|$ and $\mathbf{A}^T$ denote the determinant and transpose of a matrix $\mathbf{A}$, respectively. We now use a D-dimensional mean vector μ, and a D x D covariance matrix Σ. The quantity $(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)$ is the Mahalanobis distance from $\mathbf{x}$ to μ. We can view this as a fair distance measure in that components of $\mathbf{x}$ with large variance do not contribute as much to the distance as components with small variances. From statistical estimation theory, we know that it desirable to obtain estimates of a parameter θ, such that $\mathcal{E}[\hat{\theta}] = \theta$, where $\mathcal{E}[\cdot]$ denotes the expectation operator. Such estimates are *unbiased* estimates (31). Unbiased estimates for μ and Σ are

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^{(n)} \tag{2.11}$$

$$\hat{\Sigma} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}^{(n)}-\hat{\mu})(\mathbf{x}^{(n)}-\hat{\mu})^T, \tag{2.12}$$

where N is the number of samples and $\mathbf{x}^{(n)}$ denotes the n^{th} sample.

Let us now suppose that the class conditional densities are each Gaussian. Then for each class, we could compute $\hat{\mu}_k$ and $\hat{\Sigma}_k$ using Equations (2.11) and (2.12). Each density $p(\mathbf{x}|C_k)$ could then be expressed in the form of Equation (2.10).

Earlier we introduced discriminant functions, and monotonic transformations of these functions. In particular, we saw that by using a logarithmic transformation we obtain discriminant functions in the form of Equation (2.8). If we substitute the Gaussian form for $p(\mathbf{x}|C_k)$ in Equation (2.8), then we obtain

$$y_k(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1}(\mathbf{x} - \hat{\mu}_k) - \frac{1}{2}\ln\left|\hat{\Sigma}_k\right| + \ln P(C_k). \tag{2.13}$$

The decision boundaries are thus general quadratic surfaces due to the presence of the quadratic term $(\mathbf{x} - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1}(\mathbf{x} - \hat{\mu}_k)$. There may be simplifications to Equation (2.13). If the covariance matrices for all classes are equal, then the $\left|\hat{\Sigma}_k\right|$ terms can be dropped since they are class independent. The quadratic term $\mathbf{x}^T \hat{\Sigma}_k^{-1}\mathbf{x}$ that arises upon expansion of $(\mathbf{x} - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1}(\mathbf{x} - \hat{\mu}_k)$ also can be dropped because it too is class independent. We can then write the discriminant functions as

$$y_k(\mathbf{x}) = \mathbf{w}_k \mathbf{x} + w_{k0}, \tag{2.14}$$

where $\mathbf{w}_k^T = \hat{\mu}_k^T \hat{\Sigma}^{-1}$ and $w_{k0} = -\frac{1}{2}\hat{\mu}_k^T \hat{\Sigma}^{-1}\hat{\mu}_k + \ln P(C_k)$. The decision boundaries thus become hyperplanar .

If the features are statistically independent, then the covariance matrix is diagonal and we need only be concerned with computing the variances of the individual features. A more extreme case is when the variance is equal across all features and all classes share this variance so that $\hat{\Sigma} = \hat{\sigma}^2 \mathbf{I}$ for all classes. If this is the case Equation (2.13) simplifies to

$$y_k(\mathbf{x}) = -\frac{\|\mathbf{x} - \hat{\mu}_k\|^2}{2\hat{\sigma}^2} + \ln P(C_k). \tag{2.15}$$

In the case of equal prior probabilities, Equation (2.15) results in a simple decision rule: Measure the Euclidean distance from a feature vector to all class means and assign it to the class for which this distance is the smallest. This rule is understandably referred to as a *nearest mean classifier* (44). The mean vector plays the role of a *template* so that classification amounts to the classical technique of template matching. Templates also play a role in radial basis function classifiers.

At this point, a fair question is whether or not representing a density function as a Gaussian is a reasonable thing to do in view of its simplicity. We can look to nature for an answer to this question: It is well known that many densities arising in nature are well represented by Gaussian functions. This is true, for example, for receiver noise in electronic systems, and yearly rainfall. The central limit theorem provides a strong basis for the prevalence of Gaussian densities. This theorem states that the density function for the sum of independent random variables approaches the Gaussian form as the number of independent random variables increases without bound (48).

2.2.3 Feature Extraction and Pre-Processing.

Let us return to the problem of determining the era of a classical music composition. If you simply listen to an instrument in the composition that is faint, then you probably can not determine the era. If you listen to a lead instrument, then your chances are improved, because the lead instrument has more “presence”. Still, you should consider the overall sound, and not just the lead instrument. You can consider each instrument as a unique *feature*, or you can combine low frequency instruments and higher frequency instruments into separate features. Thus, there are many ways that you could go about *extracting* information from the composition so as to make a decision regarding its era. This process is known as feature extraction and is an extremely important step that occurs early on in the pattern recognition process.

If your classical-era decision making process is to be successful, you may need to remove noise from the recording before you extract information from it, for it may be considerably noisy as is often the case with old recordings. By removing noise, you are able to cue on the true instrumental quality of the composition. In engineering terms, you have increased the signal-to-noise ratio (SNR) so that the signal component is dominant.

In the next section, we consider wavelet-based signal analysis which can satisfy the goals of both feature extraction and pre-processing. Particular emphasis is placed on the signal denoising capabilities of wavelets. First we provide a theoretical background so that this powerful analysis tool can be appreciated.

2.3 Wavelet Analysis

The development in this section is based primarily on Chapter 2 in (8). When appropriate, it is supplemented with content from the cited references.

Often in the business of signal analysis, it is desired to express a signal $f(t)$ by a linear decomposition as

$$f(t) = \sum_{l} a_l \psi_l(t), \tag{2.16}$$

where l is an integer index, the a_l are the expansion coefficients and the $\psi_l(t)$ form the expansion set. Furthermore, we often desire that the expansion set be orthogonal so that

$$\langle \psi_k, \psi_l \rangle = \int \psi_k(t) \psi_l(t) dt = 0 \quad k \neq l, \tag{2.17}$$

where $\langle , \rangle$ denotes the inner product operation. Orthogonality then allows us to compute the a_k by

$$a_k = \langle f, \psi_k \rangle = \int f(t) \psi_k(t) dt. \tag{2.18}$$

Thus, the a_k are simply the projections of f onto the ψ_k (In fact, the a_k are the least squares projections.). Perhaps the most well known linear decomposition is the Fourier Transform (FT), which decomposes a signal into a sum of sines and cosines (or complex exponentials). A drawback of the FT is that the coefficients represent frequency components that are of an infinite duration, and so time localization is lost. If a signal contained a high frequency burst, for instance, then the Fourier representation would not tell us *when* that burst occurred. Traditional Fourier analysis, then, is not suitable for nonstationary signals (2).

To overcome the pitfalls of traditional Fourier analysis, Gabor devised a scheme which positions a window $g(t)$ at a time τ on the time axis and computes the FT of the signal within the window extent. This scheme is known as the Short-Time Fourier Transform

(STFT) and is computed as

$$F(\omega, \tau) = \int_{-\infty}^{\infty} f(t) g^*(t - \tau) e^{-j\omega t} dt. \tag{2.19}$$

When $g(t)$ is chosen to be Gaussian, then Equation (2.19) is called the Gabor transform. The problem with the STFT is that the fixed duration of the window results in a fixed frequency resolution and hence a fixed time-frequency resolution as a result of the *uncertainty principle.* Figure 2.2 shows the resolution cells that are obtained in the time frequency plane of the STFT.

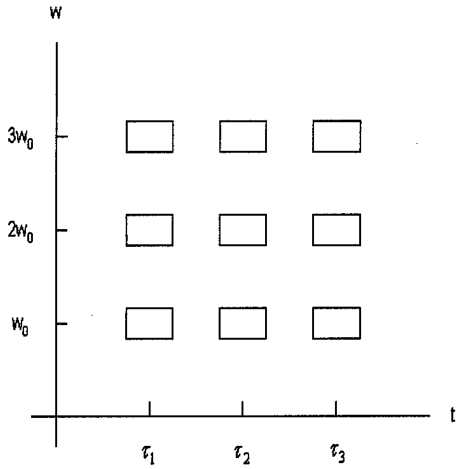


Figure 2.2 Resolution cells in time-frequency plane of short time Fourier transform.

The wavelet approach handles the drawbacks of the FT and STFT by allowing for a prototype function - the wavelet - to be scaled and shifted. A function is then projected onto the scaled and shifted versions. If we call the basic wavelet $\psi(t)$ and define $\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})$ to be the scaled and shifted version of the basic wavelet (with scale a and shift b), then we obtain the continuous wavelet transform (CWT) of a signal $f(t)$ by evaluating

$$F(a, b) = \int f(t) \psi_{a,b}(t) dt. \tag{2.20}$$

The wavelet transform thus provides us with a two-dimensional expansion set as a result of the scaling and translation operations. Depending on whether a is large or small, $\psi(t)$ either expands or contracts in time, which results in a corresponding expansion or contraction

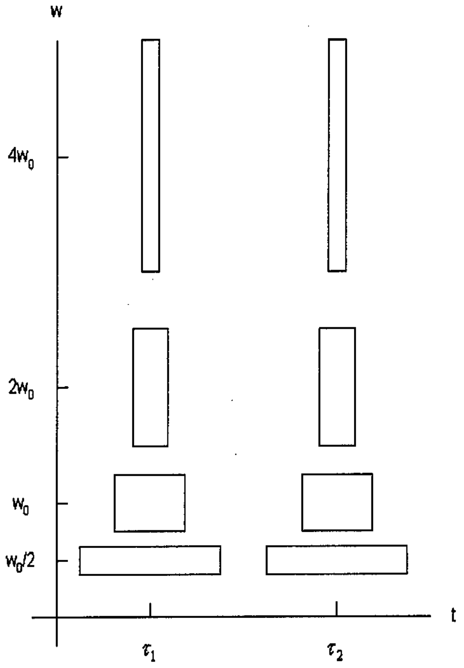


Figure 2.3 Resolution cells in time-frequency plane of wavelet transform.

in frequency. The wavelet transform then provides flexibility in time-frequency resolution. Figure 2.3 shows the resolution cells in the time-frequency plane for the wavelet transform (2).

Since the CWT maps a function of one parameter into one of two parameters, there is clearly redundancy. It turns out that we can sample the CWT plane and still get perfect reconstruction of $f(t)$ which is analogous to the result that we can recover a signal from its samples as long as the Nyquist rate is met (12). The sampling of the CWT plane is most often done on a dyadic, which means that $a = 2^j$ and $b = k2^j$. We then have $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$ (2) and can express $f(t)$ as

$$f(t) = \sum_j \sum_k d_{j,k}\psi_{j,k}(t), \tag{2.21}$$

where j and k are integer indices. Equation (2.21) provides us with a linear decomposition essentially in the form of Equation (2.16). The $d_{j,k}$ are the discrete wavelet transform (DWT) and are computed as $\langle f, \psi_{j,k} \rangle$ provided that $\{\psi_{j,k}\}$ forms an orthonormal set.

At this point one may wonder about the benefits of wavelet analysis and under what circumstances their use is advantageous. Below are some general reasons why wavelet analysis is attractive:

1. The magnitudes of the $d_{j,k}$ fall off rapidly for a large class of signals - this is a characteristic of an *unconditional basis* and makes wavelets ideal for signal compression and denoising. Donoho (22) shows that wavelets are optimal for compression and denoising for a large signal class.
2. Wavelet analysis allows for time and frequency localization so that transient features of a signal can be well represented, whereas with traditional Fourier methods there is no time localization.
3. There are many wavelets one can choose from, thereby making wavelets adaptable to a specific problem. Wavelets can be designed in a similar manner to finite impulse response (FIR) digital filters.
4. A computationally efficient algorithm exists for performing the discrete wavelet transform.

We are now ready to begin a discussion of the wavelet approach from the viewpoint of multiresolution analysis, which provides an intuitive framework on which to base much of wavelet theory. This analysis leads to a computationally efficient means for computing the wavelet transform, alluded to above, which requires $O(N)$ operations. Note that the computational efficiency exceeds that of the Fast Fourier Transform (FFT), which requires $O(N\log(N))$ operations.

2.3.1 Multiresolution Analysis. We will first describe, what is known in wavelet literature, as the scaling function. A set of scaling functions can be defined in terms of integer translates of a basic scaling function as

$$\phi_k(t) = \phi(t-k) \quad k \in \mathbf{Z}, \tag{2.22}$$

where **Z** is the set of all integers. We define the subspace spanned by these functions as

$$V_0 = \underset{k}{Span}\{\phi_k\}, \tag{2.23}$$

which means that

$$f = \textstyle\sum_k a_k \phi_k \quad \forall \quad f \in V_0. \tag{2.24}$$

We now allow the scaling function time scale to change giving us the family of functions

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k), \tag{2.25}$$

which span the subspaces

$$V_j = \underset{k}{Span}\{\phi_{j,k}\}. \tag{2.26}$$

Increasing j allows the scaling functions to represent finer detail, while decreasing j allows them to only represent coarse details - this idea corresponds to our intuition of high and low frequency in the Fourier domain.

We now establish the multiresolution framework by requiring a nesting of the spanned spaces:

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2. \tag{2.27}$$

This nesting leads to the observation that

$$f(t) \in V_j \quad \Leftrightarrow \quad f(2t) \in V_{j+1}, \tag{2.28}$$

and so we see that elements in a given space are scaled versions of elements in the next finer space. Figure 2.4 depicts the nested spaces spanned by the scaling functions. We require that $\phi(t) \in V_1$, and since it is in V_0, we can express $\phi(t)$ as a weighted sum of

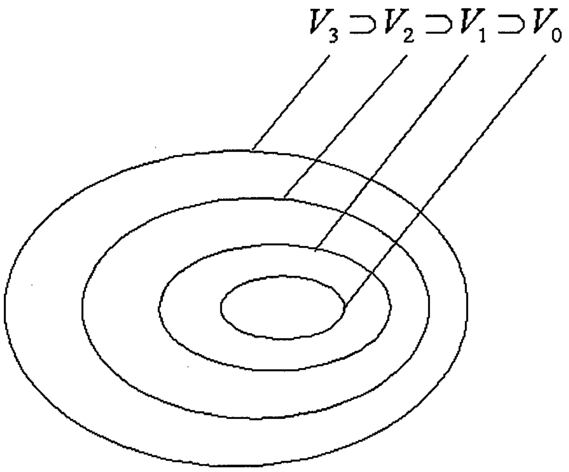


Figure 2.4 Nested spaces spanned by scaling functions.

shifted versions of $\phi(2t)$. So we have

$$\phi(t) = \sum_n g(n)\sqrt{2}\phi(2t - n), \tag{2.29}$$

where the $g(n)$'s are a real or complex sequence called the scaling filter, and $\sqrt{2}$ ensures the norm of the scaling function is one. This equation is often called the multiresolution analysis (MRA) equation.

We now define the wavelets to be functions that span the *differences* between the spaces spanned by the scaling functions, and hence the wavelets are orthogonal to the scaling functions. Figure 2.5 shows the relation among the wavelet and scaling function

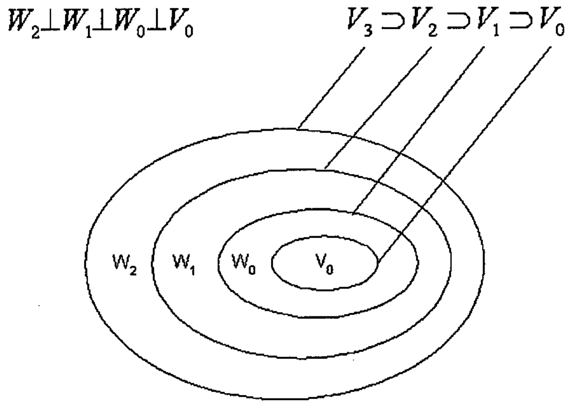

$W_2 \perp W_1 \perp W_0 \perp V_0$
$V_3 \supset V_2 \supset V_1 \supset V_0$
W_2
W_1
W_0
V_0

Figure 2.5 Nested spaces spanned by scaling functions and wavelets.

spaces. From the figure we see that

$$L^2 = V_0 \cup W_0 \cup W_1 \cup \cdots . \tag{2.30}$$

The choice for the initial space is arbitrary - it can be at higher or lower resolution than V_0. We could just as easily define

$$L^2 = V_{-5} \cup W_{-5} \cup W_{-4} \cup \cdots . \tag{2.31}$$

For practical purposes, we choose the initial space to represent the coarsest information of interest.

Figure 2.5 also shows that the wavelets are in the space of the next finer scaling function, which leads to a relation between a wavelet function and a scaling function, similar to that of Equation (2.29):

$$\psi(t) = \sum_n h(n)\sqrt{2}\phi(2t-n). \tag{2.32}$$

Equation (2.32) provides the mother wavelet, and we generate a family of wavelet functions (similar to the family of scaling functions) as

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k). \tag{2.33}$$

Now that we have the families of functions $\phi_{j,k}(t)$ and $\psi_{j,k}(t)$, we can use Equation (2.30) and define j_0 to be the starting scale to obtain

$$f(t) = \sum_{k=-\infty}^{\infty} a_{j_0}(k)\phi_k(t) + \sum_{j=j_0}^{\infty}\sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t). \tag{2.34}$$

Thus, the wavelet transform as specified by the coefficients in Equation (2.34), is a sampling of the translation and scale plane of the CWT, which results in the DWT; the form of Equation (2.34) assumes a dyadic grid. The wavelet coefficients in Equation (2.34) provide for time-frequency localization in that a coefficient associated with scale j and shift k gives information about the function f near time point $2^{-j}k$ and near frequency proportional to 2^j (38). Since the signals that we most often deal with come from sampled systems, we can discretize time and simply replace t with the discrete time variable n. We assume discrete time from here on unless otherwise stated.

The first summation in Equation (2.34) provides us with a course approximation to f which is simply the projection of f onto V_{jo}. The second summation for each j gives us finer details and are the projections of f onto the W_j spaces. It is convenient to think of the wavelet transform in terms of these vector spaces. We will define $P_{V_j}f$ as the projection of f onto the V_j vector space, and similarly $P_{W_j}f$ is the projection onto the W_j space. The expansion in terms of the projections is

$$f = P_{V_{jo}}f + \sum_j P_{W_j}f. \tag{2.35}$$

In wavelet analysis we are interested in projecting a signal onto the vector spaces in Figure 2.5 as dictated by Equation (2.35), whereas with Fourier analysis we extract information from projections of a signal onto the vector spaces spanned by sines and cosines at different frequencies. Figures 2.6 and 2.7 show a signal projected onto the scaling and wavelet function spaces.

In practice, the wavelet coefficients are computed using Mallat's algorithm (34) which arises from equations that are similar to the MRA equations for the wavelet and scaling functions. These equations are as follows:

$$a_j(k) = \sum_m h(m-2k)a_{j+1}(m). \tag{2.36}$$

$$d_j(k) = \sum_m g(m-2k)c_{j+1}(m). \tag{2.37}$$

Equations (2.36) and (2.37) tell us how to perform the DWT: Convolve the coefficients at scale j with the time reversed filter coefficients $h(-n)$ and $g(-n)$ and then downsample to get the coefficients at scale $j-1$. Figures 2.8 and 2.9 show a filter bank implementation

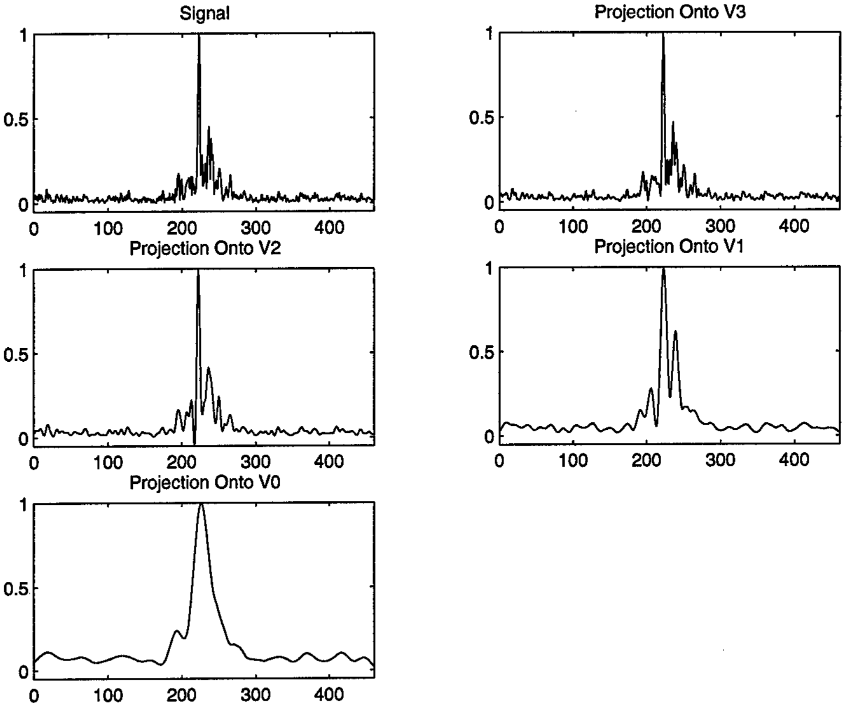


Figure 2.6 Signal projected onto the scaling function spaces.

for the decomposition and reconstruction. In fact, Mallat's algorithm can essentially be found in engineering literature on filter banks, quadrature mirror filters, conjugate filters, and perfect reconstruction filter banks. A good treatment of these subjects is in (2).

The finite signal length of sampled signals limits the number of filter bank iterations that can be carried out. Fortunately, Mallat's algorithm does provide for perfect reconstruction of sampled signals (2). We can then represent the coefficients at a given scale as a vector and express a sampled signal as

$$f(n) = R(\mathbf{a}_{j_0}) + \sum_{j} R(\mathbf{d}_j), \tag{2.38}$$

where $\mathbf{a}_{j_0}$ and $\mathbf{d}_j$ are the approximation and detail coefficients at scale j_0 and j, respectively, and $R()$ denotes the reconstruction portion that comes from either approximation or detail coefficients.

For a signal of length N Mallat's algorithm requires $O(N)$ operations. Note that there are $J-\ j_0$ *levels* in such a decomposition. The term "levels" is often used synonymously

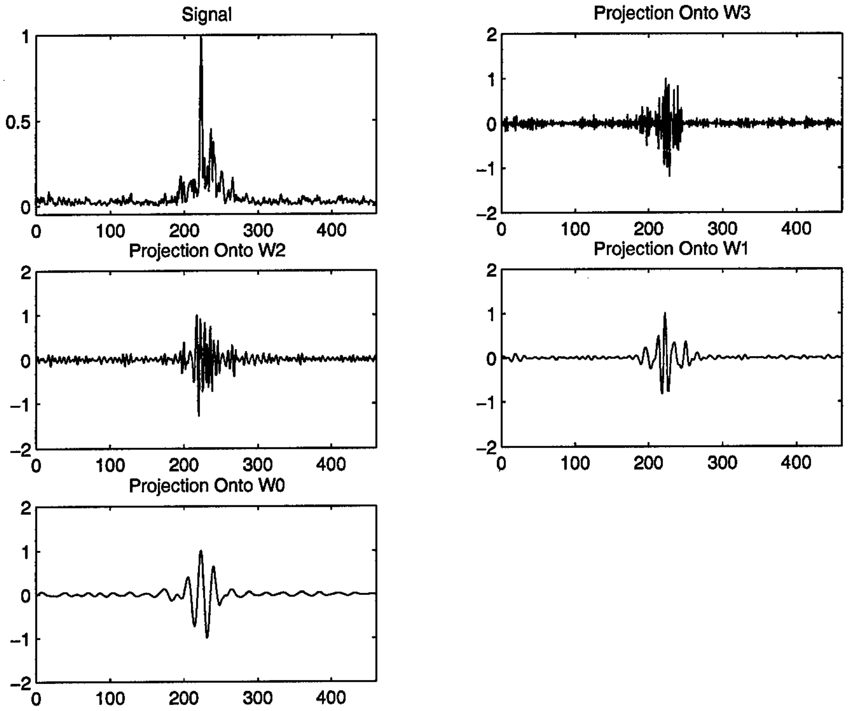


Figure 2.7 Signal projected onto the wavelet function spaces.

with *iterations.* In any case, a given level can also be labeled according to the function spaces depicted in Figure 2.5. The detail and approximation coefficients from level j provide us with the projection weights for the W_{J-j} and V_{J-j} spaces, respectively. We refer to both the levels and the spaces depending on circumstances.

As an example, consider a signal $s(n)$ whose length is $N = 2^J$, which means that there are J choices for the initial scale j_0. If we performed the full, standard wavelet decomposition as illustrated in Figure 2.8, then we would be left with a vector of wavelet coefficients which we could arrange as

$$w = [\mathbf{a}_{j_0} \mathbf{d}_{J-1} \mathbf{d}_{J-2} \cdots \mathbf{d}_{j_0}], \tag{2.39}$$

where the lengths of the coefficient vectors from left to right are $2^{j_0}, 2^{J-1}, 2^{J-2}, \ldots, 2^{j_0}$. This vector does not indicate how wavelet coefficients are typically ordered, but it is useful to arrange them in this manner for a subsequent development.

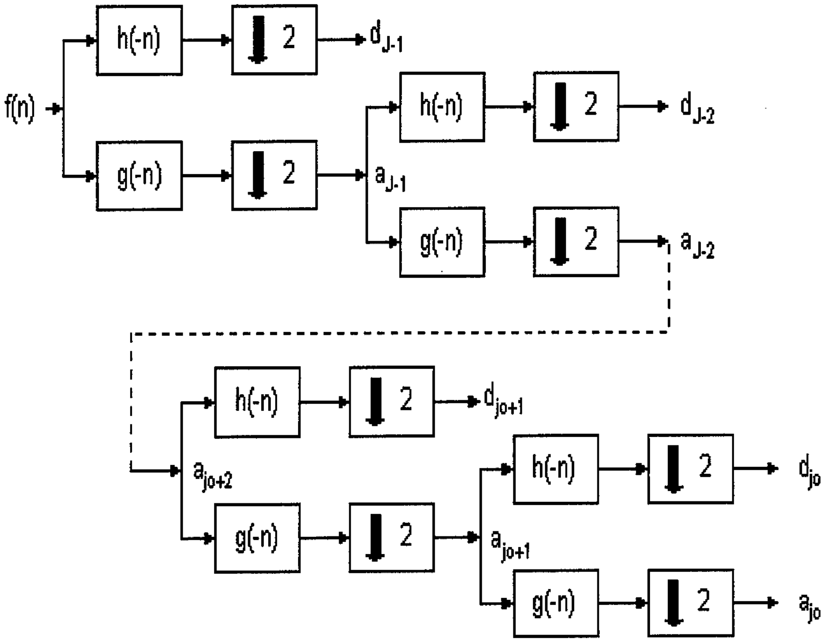


Figure 2.8 Filter bank implementation of discrete time wavelet decomposition.

The essence of designing a wavelet system amounts to determining the filters $g(n)$ and $h(n)$ that are necessitated by Equations (2.29) and (2.32). From here on we refer to these filters as the lowpass and high pass filters, respectively. The wavelet framework imposes several necessary conditions on the filters, and the design process typically involves applying digital filter design techniques using the constraints imposed on the filters. Fortunately, one does not typically need to go through this arduous process, for others have provided useful filters, with the most widely used being the Daubechies filters.

2.3.2 Daubechies Wavelets. The theory of wavelet design is beyond the scope of this thesis; (8) contains a thorough treatment of the necessary conditions imposed by wavelet systems. We summarize some important results and use them to provide insight into an important wavelet family - the Daubechies wavelets.

Design of a wavelet system can be accomplished in a manner similar to digital filter design; Equations (2.29) and (2.32) in essence define a difference equation similar to that of FIR digital filters. However, several necessary conditions arise that restrict the design

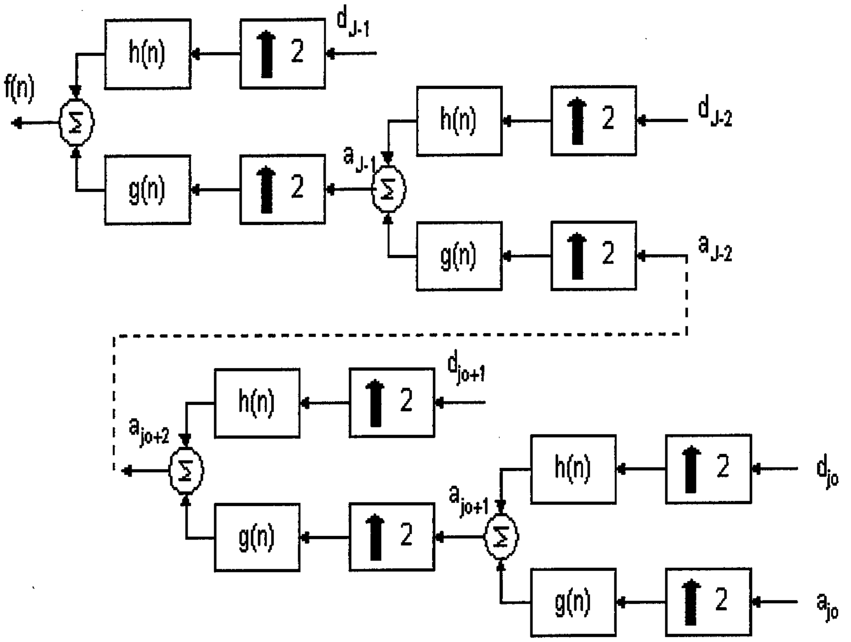


Figure 2.9 Filter bank implementation of discrete time wavelet reconstruction.

of $h(n)$. These conditions are

$$\sum_n h(n) = \sqrt{2} \quad \text{and} \quad \sum_n h(n)h(k+2m) = \delta(m). \tag{2.40}$$

The first condition is needed to ensure a solution to Equation (2.29), and the second condition provides for the orthogonality of the scaling and wavelet functions. These two conditions result in $N/2 - 1$ degrees of freedom in designing an $h(n)$ which has a length or support of N. When a Daubechies filter is specified (using the notation $daub_N$) the subscript refers to the length N.

Daubechies filters fall in an important class known as K-Regular Scaling Filters. The z-transform of such a filter is

$$H(z) = \left(\frac{1+z^{-1}}{2}\right)^K Q(z), \tag{2.41}$$

where K is the regularity of the filter and is limited by $1 \leq K \leq N/2$. The smoothness of a function is related to its differentiability and the smoothness of $h(n)$ thus is related to K, since for larger K, $|H(z)|$ drops off more rapidly. We now state some important properties of these filters:

1. The wavelet filter moments, $\mu_g(k) = \sum_n n^k g(n)$, are zero for $k = 0, 1, \ldots, (K-1)$.
2. The wavelet function moments, $m_g(k) = \int t^k \psi(t) dt$, are zero for $k = 0, 1, \ldots, (K-1)$.
3. All polynomial sequences up to degree $(K-1)$ can be expressed as a linear combination of shifted scaling filters.
4. All polynomials up to degree $(K-1)$ can be expressed as a linear combination of shifted scaling functions at any scale.

It is useful to know what order polynomials can be exactly represented in consideration of the fact that a large class of signals can be represented adequately by a truncated Taylor Series. Daubechies filters are designed by setting $K = N/2$, and so a $daub_N$ wavelet system allows us to exactly represent polynomials up to degree $N/2 - 1$ with shifted versions of the scaling functions alone. Figure 2.10 shows the wavelet and scaling functions for the $daub_4, daub_{10}$,and $daub_{16}$ wavelet systems. The Haar wavelet system is also shown for contrast in Figure 2.11.

2.3.3 Wavelet Denoising. The unconditional basis property of wavelets allows wavelets to represent a large class of signals more efficiently than, say Fourier bases. As an example, consider a piecewise constant function. The Fourier representation would require a large number of coefficients to represent the signal near discontinuities, and the basis functions would have a global extent. A wavelet representation can efficiently represent the signal at discontinuities and the basis functions have a local effect, thereby not affecting the signal representation elsewhere (38). The unconditional basis property then makes wavelets ideal for compression, and hence denoising too.

Figure 2.6 suggests a simple wavelet-based denoising method - simply retain the approximation coefficients and reconstruct the signal solely from those coefficients, i.e.,

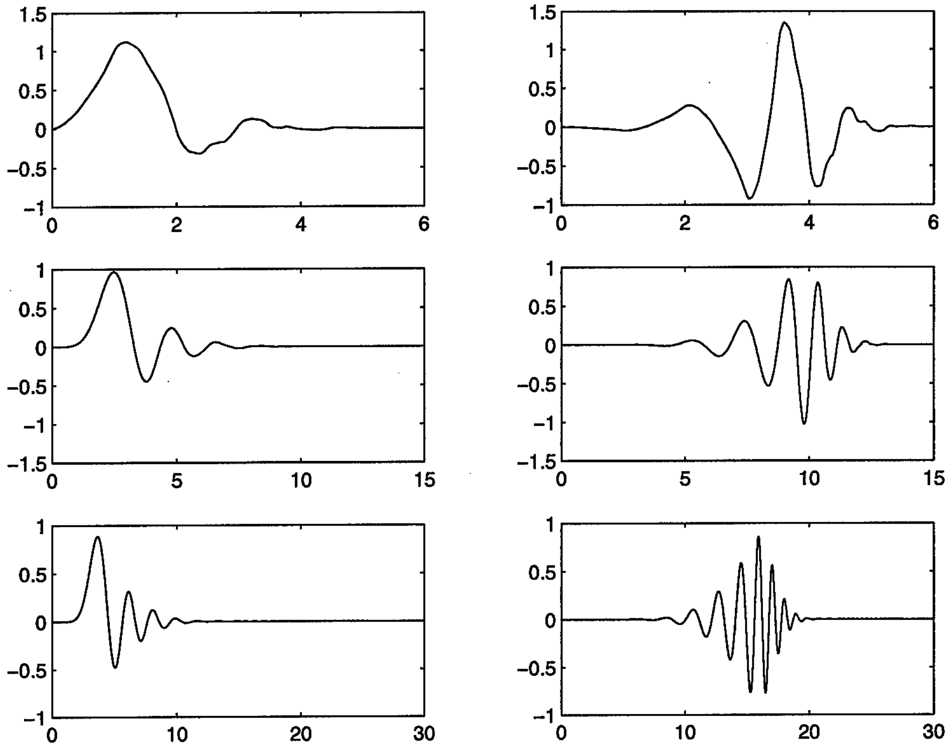


Figure 2.10 Wavelet and scaling functions for $daub_4, daub_{10}$, and $daub_{16}$.

let $P_{V_{j_o}}$ be the denoised signal. This method is crude, but it gives us a good basis for wavelet based denoising/smoothing. The more general approaches are typically established from the standpoint of nonparametric statistical estimation or regression. Donoho and Johnstone (20) take the following approach: Suppose that we have data of the form

$$s(n) = f(n) + \sigma z(n), \quad n = 0, \ldots, N \tag{2.42}$$

where the noise is independently, identically distributed as $z \overset{iid}{\sim} N(0,1)$ and σ is the noise level. By employing the usual l_2 norm, a measure of risk is defined as

$$\mathcal{R}(\hat{f}, f) = n^{-1} E \left\| \hat{f} - f \right\|_2^2 . \tag{2.43}$$

The goal then is to minimize the risk, which is done by viewing the problem as one of *selective wavelet reconstruction.* The idea is that only "large" wavelet coefficients contribute to the signal, and so to obtain the estimate $\hat{f}$ we keep only those coefficients whose magnitude

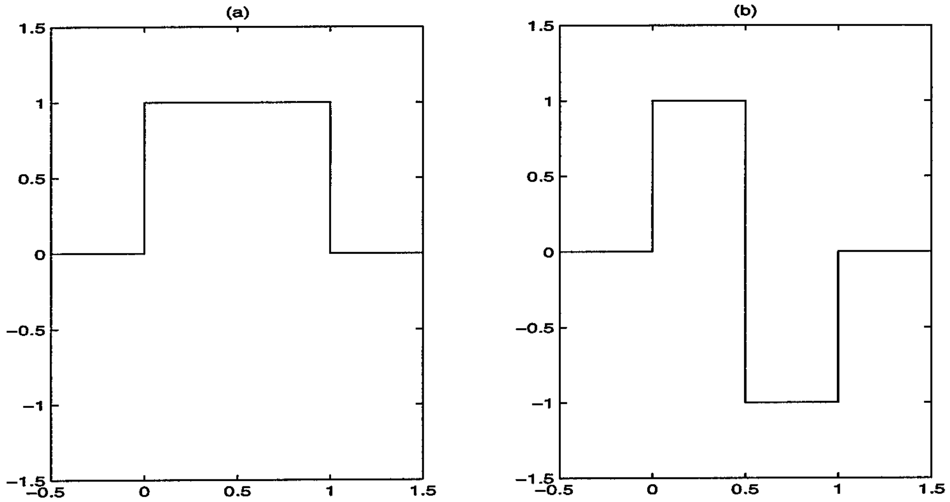


Figure 2.11 The Haar wavelet and scaling function.

are greater than a threshold t. Such a threshold scheme is known as *hard thresholding.* In recognizing that each wavelet coefficient contains a signal *and* noise portion, it is desirable to try and remove the noisy portion. Soft thresholding, like hard thresholding, aims to meet this desire by keeping only those coefficients whose magnitudes are greater than a threshold. However, the remaining coefficients are shrunk towards zero by an amount t - hence, soft thresholding is often referred to in wavelet literature as *wavelet shrinkage.* The thresholding operators are defined by Equations (2.44) and (2.45) and are illustrated in Figure 2.12.

Soft thresholding:

$$\eta_t(x) = \begin{cases} sign(x)(|x| - t) & \text{if } |x| > t \\ 0 & \text{if } |x| \leq t \end{cases} \tag{2.44}$$

Hard thresholding:

$$\eta_t(x) = \begin{cases} x & \text{if } |x| > t \\ 0 & \text{if } |x| \leq t \end{cases} \tag{2.45}$$

Both schemes possess the property of being *spatially adaptive,* meaning that they are able to apply the proper level of smoothing where needed. Donoho and Johnstone (20)

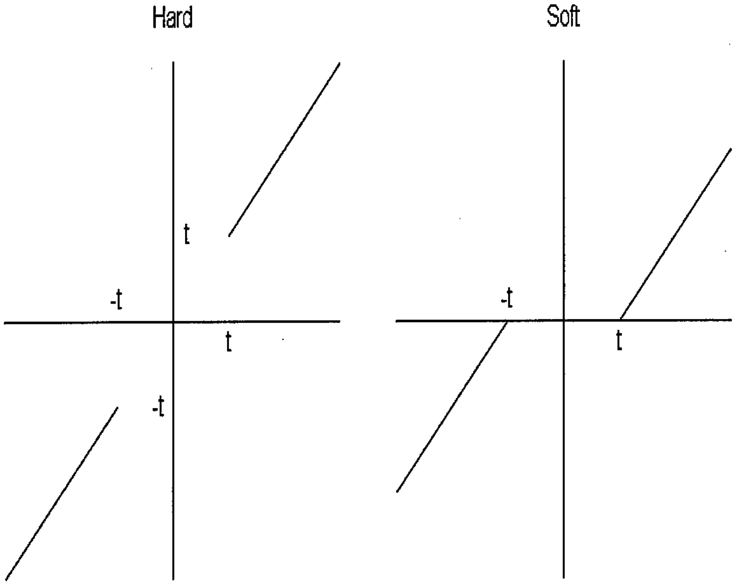


Figure 2.12 Illustration of soft and hard thresholding.

demonstrate that spatially adaptive estimation with wavelets is as powerful as other high-performance methods such as piecewise polynomial estimation. The general wavelet thresholding approach is as follows:

1. Perform decomposition of the signal using an orthogonal wavelet transform.
2. Apply soft or hard thresholding rules to the coefficients obtained in step 1.
3. Reconstruct the signal with thresholded coefficients.

In the above algorithm, the approximation coefficients are not thresholded, as they determine the underlying signal structure and contain no noise component (6).

In applying wavelet thresholding the choice of t is critical. Choosing too large a threshold results in oversmoothing, whereas choosing too small a threshold results in noisy estimates. Another issue concerns the choice of thresholding scheme. In practice, the soft thresholding method is used far more often than hard thresholding, due to the more visually pleasing results afforded with soft thresholding. Hard thresholding tends to produce greater oscillations near signal discontinuities and does not preserve signal features as well as soft thresholding (15).

One of the most widely used wavelet denoising methods is Donoho's *VisuShrink* and is described in (23). The VisuShrink threshold is set to $\sqrt{2\log(N)}\hat{\sigma}$ where $\hat{\sigma}$ is an estimate of the noise level. This is often referred to as the *universal* threshold. This choice of threshold is based on a Gaussian noise model in which case $P(\max_i |z_i| > \sqrt{2\log(N)}) \rightarrow 0$, as $n \rightarrow \infty$. Donoho suggests the use of soft-thresholding and shows that the estimate $\hat{f}$ obtained through the VisuShrink method is at least as smooth as f for a wide variety of smoothness measures and that it comes as close in mean square error to f as any measurable estimator can come. In fact, Donoho shows that when a Gaussian noise model is assumed, the noise can be *completely* removed (in an appropriate probabilistic manner) for some theoretical threshold. The details are in (23).

A well documented problem with VisuShrink is that it tends to result in oversmoothed estimates due to the fact that the threshold selection rule results in a relatively large threshold thereby shrinking or removing a large number of coefficients (35). To overcome this, Donoho and Johnstone (21), propose the *SureShrink* procedure which is based on minimizing the Stein unbiased risk estimate (SURE) and assumes the Gaussian noise model. Given N noise-corrupted wavelet coefficients $\{w_i\}$, the SURE criterion is

$$SURE(t,w) = N - 2 \cdot \#\{i : |w_i| \leq t\} + \sum_{i=1}^{N}(|w_i| \wedge t)^2, \tag{2.46}$$

where $x \wedge y$ is the minimum of x and y. An $O(N \log N)$ algorithm is used to determine the threshold that minimizes the SURE criterion. SureShrink is applied at each decomposition level and if enough signal content is available then the SURE threshold is used; otherwise the universal threshold is used for that level. A remarkable theoretical result is that the SureShrink estimate achieves the near-minimax optimal rate of convergence for estimation of a function simultaneously over a large class of functions. Kernel and spline-based methods are not able to perform in a near-minimax sense over as many function spaces as SureShrink does. This result assumes soft thresholding.

Though SureShrink alleviates the over-smoothing problem of VisuShrink, it tends to under-smooth. Neither of these methods provide desirable results in all situations. Other more general means to select the threshold are hypothesis testing and cross-validation.

Denoising in the above context is based on the assumption that there are "small" valued coefficients corresponding to noise and "large" valued coefficients corresponding to the signal. To determine these two groups of coefficients, hypothesis testing can be used so that the large coefficient group contains only coefficients that "pass the test". This approach is taken by Abramovich and Benjamini (1) and Ogden and Parzen as discussed in (39), and assumes the usual Gaussian noise model. Cross validation methods seek to choose the threshold such that the estimator has best performance when predicting new observations. A prediction is made on a subset of data and then compared to the remaining data. This data driven method makes no assumptions about the noise model. Nason uses this approach in the context of wavelets, but states that it is not suitable for complex noise structures(36), and that it is far less superior to the SureShrink method in the case of correlated noise (38).

Ghael recognized that for any given signal, the optimal denoising method with respect to mean squared error (MSE) is the Wiener filter. However, Wiener filtering requires knowledge of the signal and noise statistics. A wavelet shrinkage estimate can be used as a preliminary step in designing a wavelet domain Wiener filter. This technique is aptly named *WienerShrink.* Whereas denoising methods typically strive to balance variance and bias, the WienerShrink method improves both simultaneously (27). Donoho points out that optimality with respect to mean square error alone tends to result in undesirable side effects - "ripples," "blips," and oscillations. VisuShrink is subject to an additional condition that with high probability, $\hat{f}$ is at least as smooth as f. It is this extra condition which allows VisuShrink estimates to be more visually appealing than other methods. This visually pleasing quality is responsible for the coining of the term "VisuShrink."

One drawback with the above denoising approaches stems from the fact the wavelet transform is not translation invariant. The set of coefficients in Equation (2.39) would not be the same for a signal and a shifted version of that signal, and there is no simple relationship between the two sets (8). We prefer the denoising process to be translation invariant in the following sense: Consider a function f. Denote $S_s f$ to be a circularly shifted version of f. Also let $\hat{f}$ and $\hat{f}_s$ be the denoised versions of f and $S_s f$, respectively. We want $\hat{f}_s$ to simply be a shifted version of $\hat{f}$, so that $\hat{f}_s = S_s \hat{f}$.

Several methods have been developed to perform translation invariant denoising. Coifman (15) developed one such method termed *cycle spinning.* In this approach, a signal is shifted, denoised, then unshifted and averaged across the shifts. When cycle spinning is done over all shifts the translation dependence is averaged out. The interest in cycle spinning led to a computationally efficient algorithm which, remarkably, requires $\mathcal{O}(N\log(N))$ time to cycle over *all* shifts. It turns out that we can in essence perform cycle spinning over all shifts, but we do not need to actually perform all shifts. This fact was realized by Beylkin and others independently (8). First, consider either the high pass or low pass output of the first stage in Figure 2.8. The decimation process discards the odd indexed coefficients, and leaves the even indexed coefficients. Now suppose the input signal is shifted by one, which shifts the coefficients by one as well, and so the odd indexed coefficients remain, and the even indexed coefficients are discarded. Thus the set of coefficients to be further processed are completely different for the two cases. However, if we were to shift the input signal by two, then the decimated output would differ from the nonshifted output by a shift of one. So all even shifts result in the *same* coefficients - they are just shifted versions of one another and the same is true for all odd shifts. At the end of the first iteration of Figure 2.8, there are a total of N different detail coefficients, where N is the signal length. These coefficients are split into two groups, each with $N/2$ coefficients. One group corresponds to the nonshifted output and the second group corresponds to the output when the input is shifted by one. The approximation coefficients are also split into two groups in a similar manner. These two approximation groups are further processed as the decomposition goes through the next iteration. By the argument above, each of these two groups, when processed at the next level, spawns two more unique groups of approximation and detail coefficients. The following algorithm now emerges:

1. Apply the filtering block of Figure 2.13 to the input signal - S_1 signifies a shift of one.
2. Keep the groups of detail coefficients and proceed to next level of decomposition.
3. Apply the filtering block of Figure 2.13 to all groups of approximation coefficients outputted from step 2.

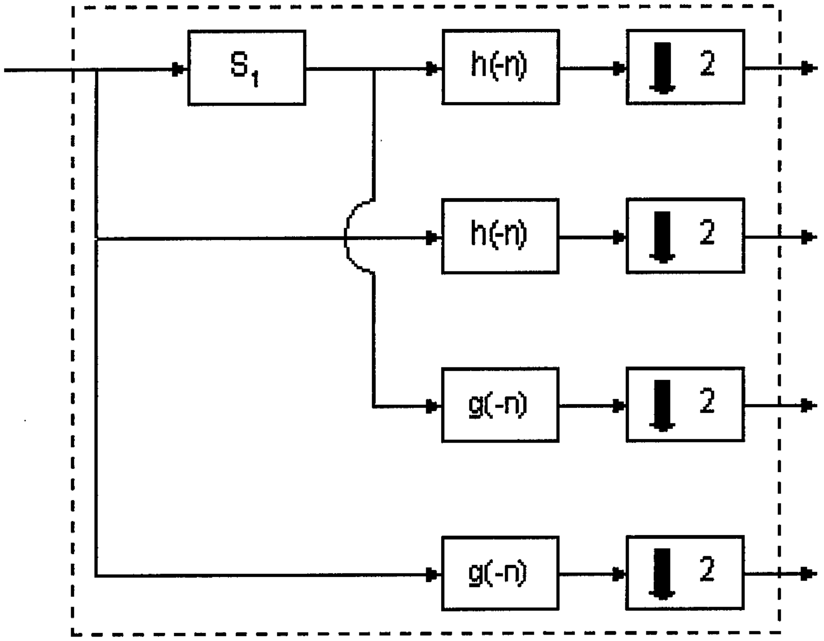


Figure 2.13 Filter block for TI wavelet transform implementation.

4. Repeat steps 2 and 3 until the coarsest level of the decomposition has been completed.

The above algorithm requires $\mathcal{O}(N\log(N))$ time, and keeping track of the groups of coefficients is a major bookkeeping issue. To handle bookkeeping, Donoho and Coifman (15) use a data structure referred to as the Translation Invariant (TI) Table. The TI Table is an array of size N X $J - j_0 + 1$, where $N = 2^J$ is the signal length and j_0 corresponds to the coarsest scaling function space - V_{j_0}. The first column contains $N/2^{j0}$ groups, each having 2^{j0} elements. The groups correspond to the unique collections of approximation coefficients that can result after the completion of the above algorithm. The remaining columns from left to right contain the groups of detail coefficients that result each time step 2 is completed in the above algorithm. If we number these columns from 1 to $J - j_0$, (consider the column of approximation coefficients column 0), then column k corresponds to the k^{th} iteration of the decomposition, and in it are 2^k groups of detail coefficients, each group containing 2^{J-k} coefficients. To summarize the TI Table structure, column zero contains all collections of approximation coefficients at the coarsest scale, column two contains all collections of detail coefficients at the finest scale, column three contains all collections of detail coefficients at the next finest scale, etc. The last column, then, contains all collections of detail coefficients at the coarsest scale, and hence it consists of the same number of collections as there are in the first column.

If we denote $\mathbf{d}_{j,n}$ as the n^{th} group of detail coefficients corresponding to the W_j vector space which occur at level $J-j$, and define $\mathbf{a}_{j,n}$ similarly for the approximation coefficients, then the TI table is as follows:

$$TI = \begin{pmatrix} \mathbf{a}_{j_0,1} & \mathbf{d}_{J-1,1} & \mathbf{d}_{J-2,1} & \cdots & \mathbf{d}_{j_0,1} \\ \mathbf{a}_{j_0,2} & \mathbf{d}_{J-1,2} & \mathbf{d}_{J-2,2} & \cdots & \mathbf{d}_{j_0,2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{a}_{j_0,2^{J-j_0}} & \mathbf{d}_{J-1,2} & \mathbf{d}_{J-2,4} & \cdots & \mathbf{d}_{j_0,2^{J-j_0}} \end{pmatrix} \tag{2.47}$$

Note that each $\mathbf{a}_{j,n}$ and $\mathbf{d}_{j,n}$ is a column vector with 2^{J-n} elements.

The translation invariance property of the TI table comes about because if we let $TI(f)$ be the TI table for a signal f and we let $TI(S_s f)$ be the TI table for a shifted version of f, then there is a permutation of matrix entries Π_s so that $\Pi_s TI(f) = TI(S_s f)$. The coefficients of the standard wavelet transform of f are contained in the TI table; they are the top-most group of coefficients in each column. To extract the standard wavelet coefficients for $S_s f$, Donoho and Coifman specify an encoding of the shift s and then use a dynamic programming algorithm to perform the extraction.

Of course, one needs to be able to invert the TI transform, and to do so the following algorithm is used:

1. Start with $j = j_0$.
2. For each k in the range $1 \leq k \leq 2^j$, compute $\gamma_k = (\mathbf{G}\mathbf{a}_{j,2k-1} + S_{-1}\mathbf{G}\mathbf{a}_{j,2k})/2$ and $\delta_k = (\mathbf{H}\mathbf{d}_{j,2k-1} + S_{-1}\mathbf{H}\mathbf{d}_{j,2k})/2$. $\mathbf{G}$ and $\mathbf{H}$ correspond to the usual upsampling and filtering operations used in wavelet reconstruction.
3. Compute $\mathbf{a}_{j+1,k+1} = \gamma_k + \delta_k$.
4. Set $j = j+1$ and repeat steps 2 and 3. Once you reach $j = J$, stop.
5. Set $f = \mathbf{a}_{J,1}$.

This algorithm amounts to an average of all N reconstructions from all N circulant shifts. This result is due to the fact that each γ_k and δ_k is an average of two possible reconstructions - one from an unshifted group and one from a shifted group.

We can extend the idea of the projections of Equation (2.35) to the TI wavelet transform. The projections now become averages of projections across a range of circulant shifts. Thus we can define $\bar{P}_{V_j} f$ to be the average of the approximation space projections at scale j of f for a given range of shifts, and $\bar{P}_{W_j} f$ is defined analogously. We can define $\bar{R}(\mathbf{a}_j)$ and $\bar{R}(\mathbf{d}_j)$ similarly, where $\mathbf{a}_j$ is the first column of the TI Table and $\mathbf{d}_j$ is a vector composed of the $\mathbf{d}_{j,n}$ occurring in a given detail column of the TI Table. A signal can then be represented conceptually as

$$f = \bar{P}_{V_{j_o}} f + \sum_j \bar{P}_{W_j} f \tag{2.48}$$

or

$$f(n) = \bar{R}(\mathbf{a}_{j_0}) + \sum_j \bar{R}(\mathbf{d}_j). \tag{2.49}$$

Donoho and Coifman (15) arrived at an interesting theoretical result regarding use of the Haar wavelet in the full TI denoising scheme. They showed that $\bar{P}_{V_j} f$ is a continuous function which justifies the use of the Haar wavelet when used in the full TI wavelet transform, in contrast to the case of the traditional wavelet transform, in which the Haar wavelet is used typically for illustrative purposes.

To perform translation invariant denoising, Donoho and Coifman (15) suggest thresholding the columns of detail coefficients in the TI table and then inverting to obtain the denoised signal. Denoising in this way has several advantages. To see why, note that traditional wavelet denoising results in Gibbs phenomena. Though this effect is not as serious as with Fourier-based denoising, suppression is still desirable. Gibbs artifacts can all be attributed to misalignments of features of the signal and features of the basis functions. Other TI denoising schemes attempt to find an optimal shift for the input signal which handles the feature alignment problem. These schemes consider the transforms of the N

different circulant shifts as being N transforms into N orthogonal bases. Finding the best shift can be done in a manner similar to the entropy-based best basis algorithms discussed in (14). However, when a signal contains several discontinuities, the best shift for one discontinuity may be the worst for another. The TI denoising scheme described above avoids this potential interference problem by averaging over all shifts. A method that is equivalent to the full TI denoising method is denoising using the stationary or undecimated wavelet transform. Donoho and Coifman point out this equivalence (15), and the details are found in (37).

2.4 Summary

In this chapter we present the necessary theory that forms the basis for this thesis. The theory behind the Gaussian classifier and the interpretation of the classifier reducing to a template matcher in the case of disregarding the variance information is of much interest. Also of particular interest is the conceptual signal representation afforded by wavelet analysis - that is, the summations of the vector space projections. In the case of the translation invariant wavelet transform, these projections become averages, and composing these averaged projections is done through use of a data structure that conveniently arranges the wavelet coefficients. In the following chapter, we present a wavelet-based denoising scheme that is integrated into a Gaussian classifier as a pre-processing step.

III. Methodology

3.1 Introduction

A primary objective of this thesis is to achieve an improvement in the classification accuracy of HRR data. We have seen that typical HRR signatures contain jagged features which result from the high range resolution ability of the radar. So let us suppose that a considerable portion of the signal has no discriminatory content, and that this content is a source of performance degradation. This claim is valid in view of the results obtained by MacDonald (33) and Eisenbies (24). MacDonald demonstrated that using as few as five low frequency Fourier components from an HRR signature yielded an improvement over the baseline classifier. Eisenbies improved classification by retaining as few as 5% of the range bin features. Both results lead to the same conclusion - discarding HRR signal information is advantageous. The discarding of information is an act of simplification, and by virtue of Occam's razor, we should favor such simplifications. A contrary view is that information removal is detrimental because classification performance is optimal when using the raw data (26). This view does not take into account the fact that raw data often has a low SNR and that the removal of noise can increase the SNR and hence lead to classification improvement.

The approach taken here is to simply perform a pre-processing step and then perform classification using the baseline Gaussian classifier. We have little reason to suspect that an alternative classifier alone will produce the classification improvement that we seek. The HRR range bins can be transformed so that they are governed by Gaussian probability densities, and so the Gaussian classifier is optimal under the Bayesian framework discussed in Chapter 2. Perhaps there is a yet-to-be-discovered feature set that would lend itself to non-parametric classification, but in the absence of such a feature set, we are justified in implementing a Gaussian classifier. The block diagram in Figure 3.1 depicts the baseline classification scheme and shows how an additional pre-processing step is added. Before continuing, we examine the Gaussian classifier as used in classifying HRR signals.

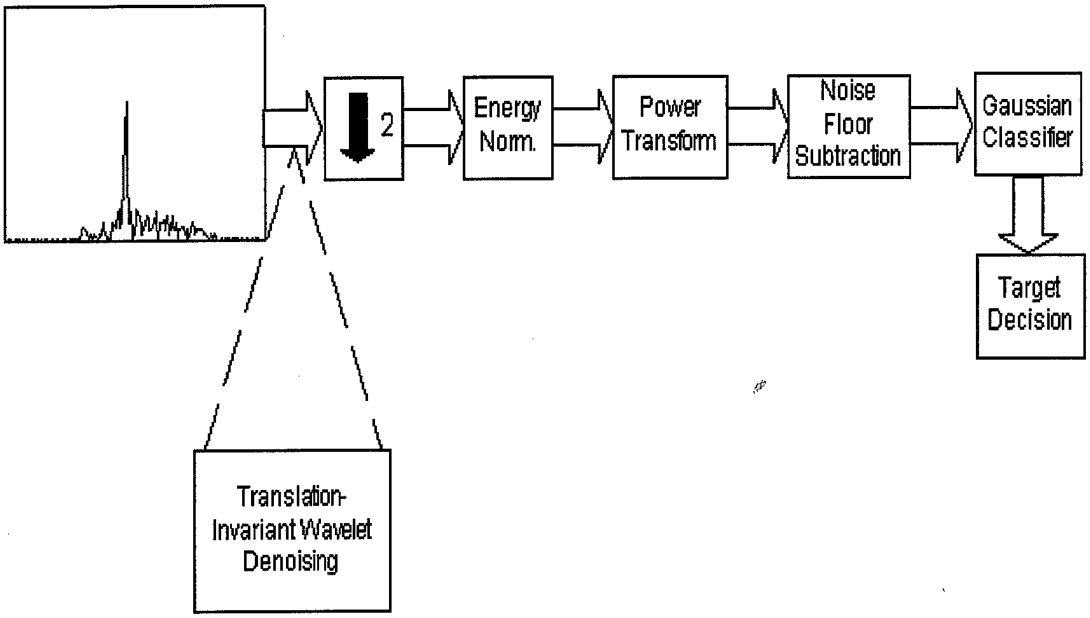


Figure 3.1 Block diagram of baseline classifier with denoising process indicated.

3.2 Baseline Gaussian Classifier

The next sections cover all the training and testing steps used by the classifier as depicted in Figure 3.1. In Chapter 2, the Gaussian classifier discriminant was given as:

$$y_k(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1}(\mathbf{x} - \hat{\mu}_k) - \frac{1}{2}\ln\left|\hat{\Sigma}_k\right| + \ln P(w_k). \tag{3.1}$$

We assume equal prior probabilities so that the last term in Equation (3.1) vanishes. Zumwalt (54) found that with limited training data, variance information can be discarded with no significant effect on classifier performance; this was true for measured and synthetic training data. Equation (3.1) then degenerates to

$$y_k(\mathbf{x}) = -\left\|\mathbf{x} - \hat{\mu}_k\right\|^2, \tag{3.2}$$

and so we are performing classification based on Euclidean distance among feature vectors and templates. Dewall (19) attributes Zumwalt's results to an insufficient number of training signatures which results in inaccurate variance estimates. Larger data sets surely warrant the use of the variances, since estimation accuracy depends on the sample size.

For the case of synthetic data, variances are ignored regardless of the number of training signatures since the noise process responsible for measured signature variance is not present in synthetic data.

3.2.1 Classifier Training.

The training of a Gaussian classifier amounts to determining the parameters of the discriminant functions. Thus, we simply estimate the mean vector $\hat{\mu}_k$ for each target. A decision first has to be made as to which signatures are used for training. After selecting the training data, several pre-processing steps are taken. After completion of pre-processing, the templates are formed. These steps are shown in Figure 3.1 and are described below.

3.2.1.1 Pre-processing.

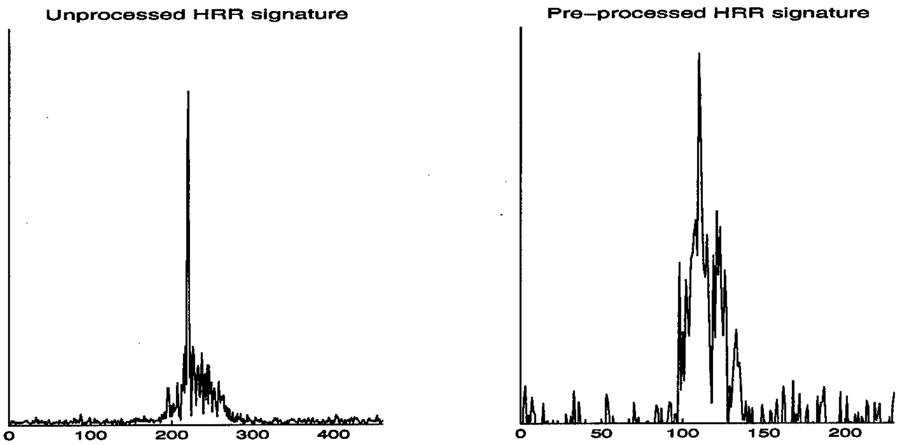
First, the signal is decimated by a half, which serves as crude dimensionality reduction and results in 230 remaining range bin features. The signal is then energy normalized, which is a necessary step since template matching amounts to classifying a signature based on the most similar template. Normalization places two signatures acquired at different ranges on the same comparative basis. Power normalization is done by first computing the signal power as

$$P = \sqrt{\frac{\sum_{i=1}^{230} s(i)^2}{230}} \tag{3.3}$$

and then dividing the signal by P. It is known that the underlying probability density of the HRR signatures is Rician (19), and so the signatures must be modified such that the densities become more Gaussian-like. This step can be performed by a power transform, which is a transformation of a random variable of the form $Y = X^v$, with $0 < v < 1$ (26). Zumwalt and Eisenbies found that $v = 0.4$ led to best classification results (54, 24). The last pre-processing step is noise floor removal. This step involves computing the mean of the first 20 range bins and subtracting that value from the entire signal. Figure 3.2 shows an example of a raw HRR signature and the pre-processed version of the same signature.

3.2.1.2 Template Construction.

After pre-processing, training signatures are averaged to form a template, which is done for each class in the problem set. For


Unprocessed HRR signature

Pre–processed HRR signature

Figure 3.2 Unprocessed HRR signature (left) and pre-processed HRR signature (right).

the case of a six class problem, Figure 3.3 shows the templates. Note that the averaging process is a low pass filtering operation.

3.2.2 Classifier Testing. An unknown signature must go through the same pre-processing steps as the training signatures, unless the training data is synthetic in which case noise floor removal is not performed. It can not be classified at this point, however, due to a well documented problem encountered when classifying HRR signatures using template matching: From class to class, there is typically a misalignment of the signature range bins. Thus, an unknown signal must be aligned with each template so that a meaningful distance computation can be performed. The alignment may be accomplished by finding the maximum of the cross correlation function of the unknown signal and the template. Once alignment is performed, distance measures are established between the unknown signal and all the templates. Figure 3.4 illustrates this process. Classification is then a simple matter of assigning the target to the class for which the distance measure is the smallest.

3.3 Wavelet Denoising

We have seen how a HRR signature is characterized by peaks that correspond to major scatterers on a target. A desirable pre-processing step is to remove the noise that is inevitably introduced by the atmosphere and by electronic systems used to process

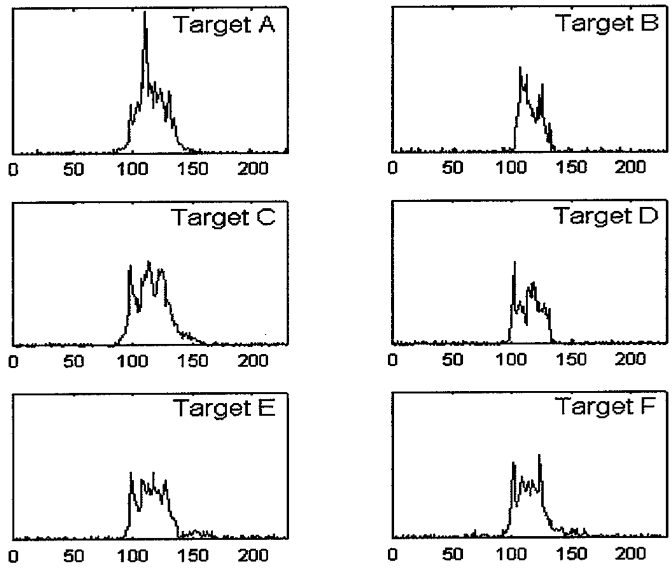


Figure 3.3 Templates for all targets after pre-processing.

the radar returns. Noise removal, regardless of the particular implementation, typically requires some assumptions about the noise. Many well-established methods assume a given probabilistic noise model and are optimized to remove the noise in the probabilistic framework. Removal of noise thus results in an estimation of the signal of interest. We use a wavelet-based approach for denoising.

Wavelet denoising has been successfully applied in many areas including EEG signals for evoked response identification (10), underwater acoustic signals (47), transient detection in noisy time series (9), and ultrasound data for feature preservation (11) just to name a few. The technical literature does not consider wavelet denoising (and denoising in general) as applied to HRR classification. Instead, wavelets have been used for feature extraction, in which case the decompositions themselves are used for classification (53, 3, 24) or some derived feature such as energies computed from the coefficients are used(49). We take a much different approach, in that the use of wavelets is solely for denoising, and no wavelet based feature extraction is performed.

Recall from Chapter 2 that a starting point for the development of most denoising techniques is a signal model in the form of Equation (2.42). Some measure of risk is defined and minimized (such as that of Equation (2.43)). If the noise model is assumed to

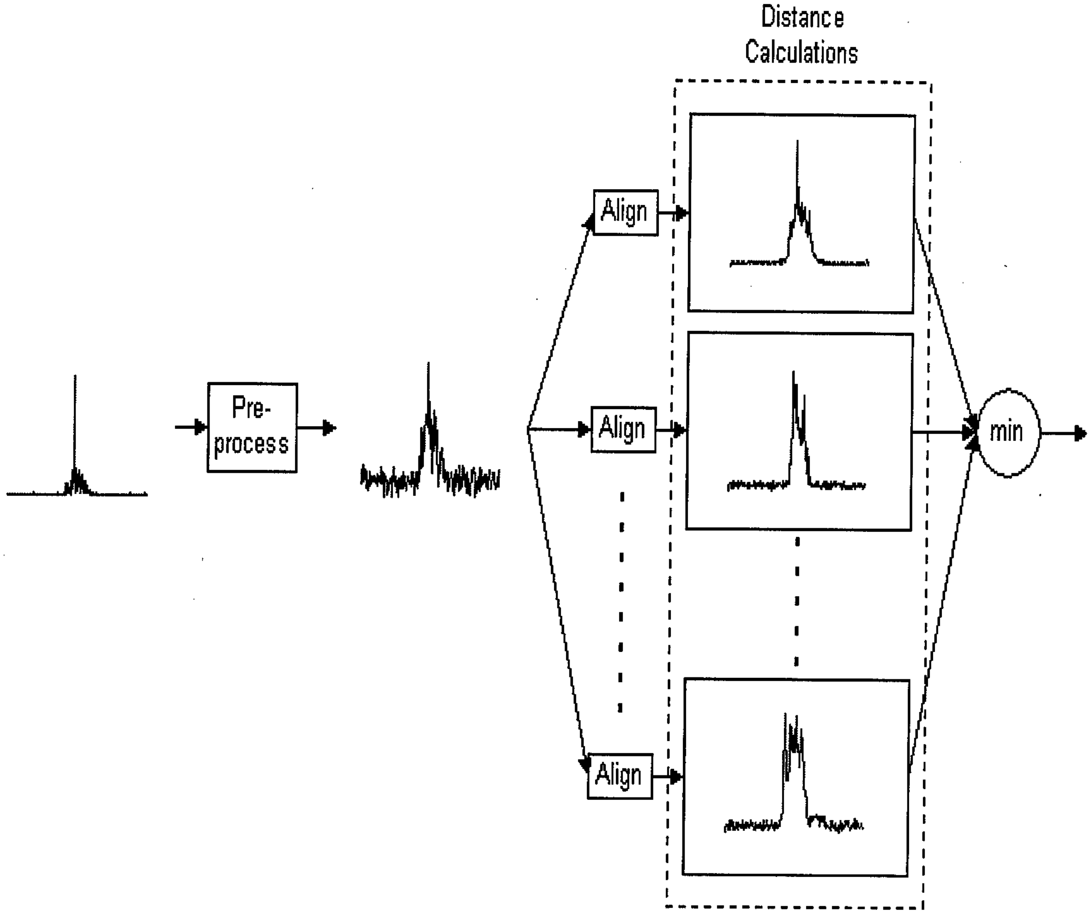


Figure 3.4 Illustration of the testing process of the baseline classifier.

be Gaussian, analytical properties can be derived, such as bounds on the risk and metrics regarding optimality relative to some criterion. For instance, Donoho's techniques can be shown to be near optimal in the minimax sense (23, 20). If one were solely interested in removing Gaussian noise, then the VisuShrink method of Donoho would be extremely attractive. Since the HRR signatures are processed by electrical systems, we may be justified in assuming a Gaussian noise model. However, we make the assumption that other forms of "noise" are present as well. We suppose that all forms of noise combine in some manner (additive, multiplicative or both) that leaves the resulting noise model unknown. The signal model then has the form

$$s(n) = f(n) + z(n), \quad n = 0, \ldots, N. \tag{3.4}$$

Here $z(n)$ is noise in an abstract sense and $f(n)$ is a representation of a HRR signal that lends itself to classification. The HRR signature fluctuations that occur due to small changes in target orientation can be considered "noise", as can the creeping wave reflections and resonance effects mentioned in Chapter 1. More importantly, the qualities of the signature that inhibit the classifier from performing at its best are "noise".

We are not interested in minimizing a risk as in Equation (2.43). We assume that since we know nothing about the noise, we do not know the form of the recovered function. In essence, what we do is not so much signal denoising as signal transformation. Our interest is in being able to transform the HRR signals so that when presented to a classifier, an improvement in classification results. The denoising scheme must be general enough so that it allows for a large class of signal realizations, because we have no knowledge of what signal forms result in classification improvement. We do not know how rough or how smooth they are but we want to make sure that both rough and smooth realizations are allowed. The denoising scheme must be optimized with respect to classification accuracy so that the optimizing procedure reveals the form of f in Equation (3.4). Since we are optimizing for classification accuracy, the optimal threshold selection techniques in the wavelet literature are not of value.

The translation invariant wavelet transform introduced in Chapter 2 serves as the engine for our denoising scheme and is computed using the third-party Matlab toolbox Wavelab, developed by Donoho and colleagues (7). Intuitively it is sensible that the denoising of a HRR signature should be independent of time (or range bin) origin. Denoising in the TI sense then supports this intuition. We repeat here the structure of the TI table:

$$TI = \begin{pmatrix} \mathbf{a}_{j_0,1} & \mathbf{d}_{J-1,1} & \mathbf{d}_{J-2,1} & \cdots & \mathbf{d}_{j_0,1} \\ \mathbf{a}_{j_0,2} & \mathbf{d}_{J-1,2} & \mathbf{d}_{J-2,2} & \cdots & \mathbf{d}_{j_0,2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{a}_{j_0,2^{J-j_0}} & \mathbf{d}_{J-1,2} & \mathbf{d}_{J-2,4} & \cdots & \mathbf{d}_{j_0,2^{J-j_0}} \end{pmatrix} \tag{3.5}$$

Before we begin to formulate a denoising scheme, we must consider the following questions:

1. What wavelet family are we to use?
2. How do we choose the coarsest scale of the wavelet decomposition?
3. Which thresholding method do we choose - soft or hard?
4. How do we choose a threshold?
5. How do we apply a threshold to the TI table?
6. How do we optimize?

These questions help define the parameters of our denoising scheme. We will address each question individually.

3.3.1 Wavelet Selection. Wavelet selection is not often discussed in the literature. For most practical purposes any orthogonal wavelet suffices except the Haar wavelet (due to the discontinuity of the wavelet and scaling function). Perhaps the most commonly used wavelet family is the Daubechies wavelets. Weiss and Dixon (47), for example, selected a $daub_4$ wavelet for denoising purposes and admitted that wavelet selection was not optimized and that additional performance gains would likely result from optimizing the wavelet selection.

In Chapter 2 we saw that the Daubechies filters fall under the class of K-Regular Scaling Filters and that they possess some important properties. For a $daub_N$ scaling filter a large N corresponds to a larger degree of smoothness. Also, large N enables exact representation of higher order polynomials by linear combinations of shifted scaling filters. Since we assume no knowledge of the function f (although we certainly do hope that f is smooth), we have no reason to choose any one Daubechies wavelet over another. Therefore we try Daubechies filters for $N = 4, 6, \ldots, 16$. We also consider the *Haar* wavelet, which would surely be unwise if we were using the traditional wavelet transform. However, as pointed out in Chapter 2, the *Haar* wavelet can provide favorable results when used in a fully translation invariant scheme.

3.3.2 Choice of Coarsest Scale. Choosing one possible coarsest scale would not give us the flexibility that we seek. Instead, we consider the projections of an HRR signature onto the scaling function spaces. We compute the projections and display them in a manner similar to Figure 2.6. However, in Chapter 2 we discussed projections in the context of the translation invariant wavelet transform, and so we are interested in visually examining the $\bar{P}_{V_j} f$ projections in the case of full translation invariance. Although we mentioned that no assumptions are made regarding the form of the denoised signals, a clarification is that we certainly expect some degree of peak information to be preserved. The scale before which prominent peak information is essentially lost is the scale that will set the lower bound for j_0. We choose the Haar basis for these projections because of its simplicity.

Figures 3.5 and 3.6 show the projections for a representative measured HRR signature for target A. We see that at V_5 the relative peak information has essentially been lost. From this observation we decide to consider j_0 in the range $6 \leq j_0 \leq 8$. We can view j_0 as a smoothing parameter (as seen in the Figures 3.5 and 3.6).

3.3.3 Hard or Soft Thresholding. If we were performing traditional wavelet denoising, then we could, with confidence, disregard hard thresholding, for it tends to produce greater oscillations near discontinuities than does soft thresholding. However, hard thresholding can not be disregarded with TI denoising because the averaging that occurs in TI denoising damps out the discontinuities of hard thresholding. Donoho and Coifman found that, in general, hard thresholding combined with translation invariance leads to superior visual and quantitative characteristics (23). More surprisingly, it was the Haar wavelet that lead to the best results! In an application involving wavelet denoising of ultrasound data, it was also found that hard thresholding along with translation invariance lead to

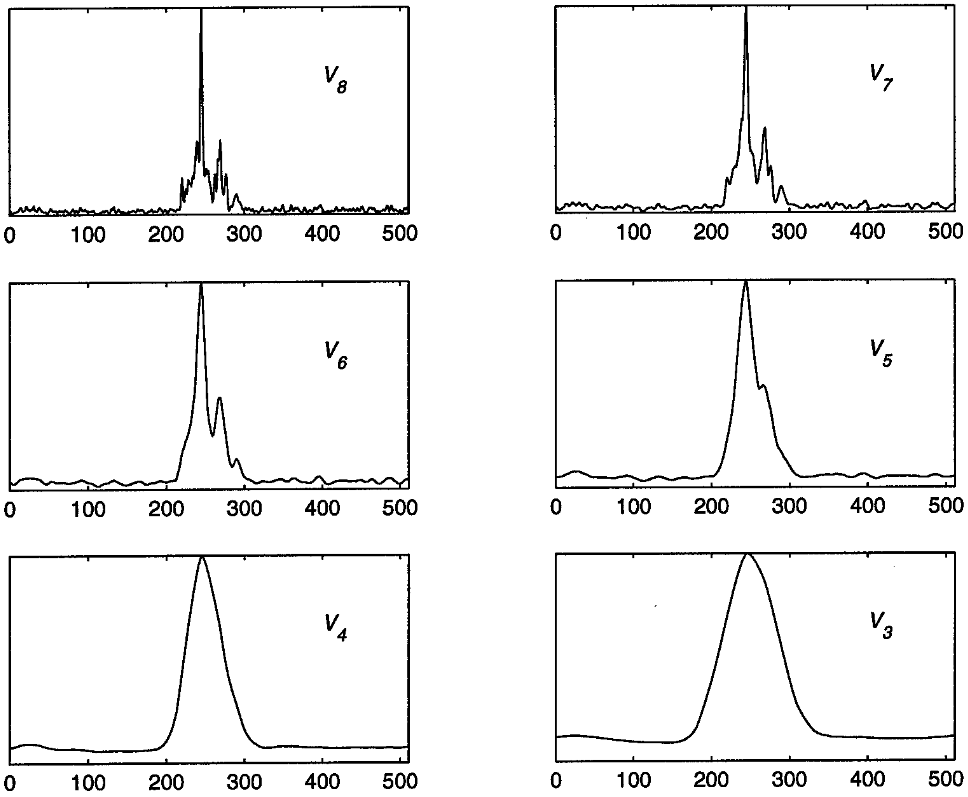


Figure 3.5 Fully translation invariant projection of measured HRR signature onto scaling function spaces.

improved performance (11). While these results do not justify ignoring soft thresholding, they do justify considering both threshold methods.

3.3.4 Applying Thresholds to TI Table. One of the trade-offs in choosing TI denoising over traditional denoising is the fact that we have more decisions to make due to the added information provided by the TI table. We must decide how to apply thresholds to the table. It is desirable to apply the thresholds so that the process is adaptive - that is, we do not want a method that results in discarding a fixed number of coefficients for all signals.

We define $\eta_t(\mathbf{c})$ to be the thresholding operator that applies a threshold t to the coefficients contained in the vector $\mathbf{c}$. This operator returns the thresholded vector $\mathbf{c}_t$. It is convenient to restrict t to the range $0 \leq t \leq 1$. The thresholds are then *relative* to a

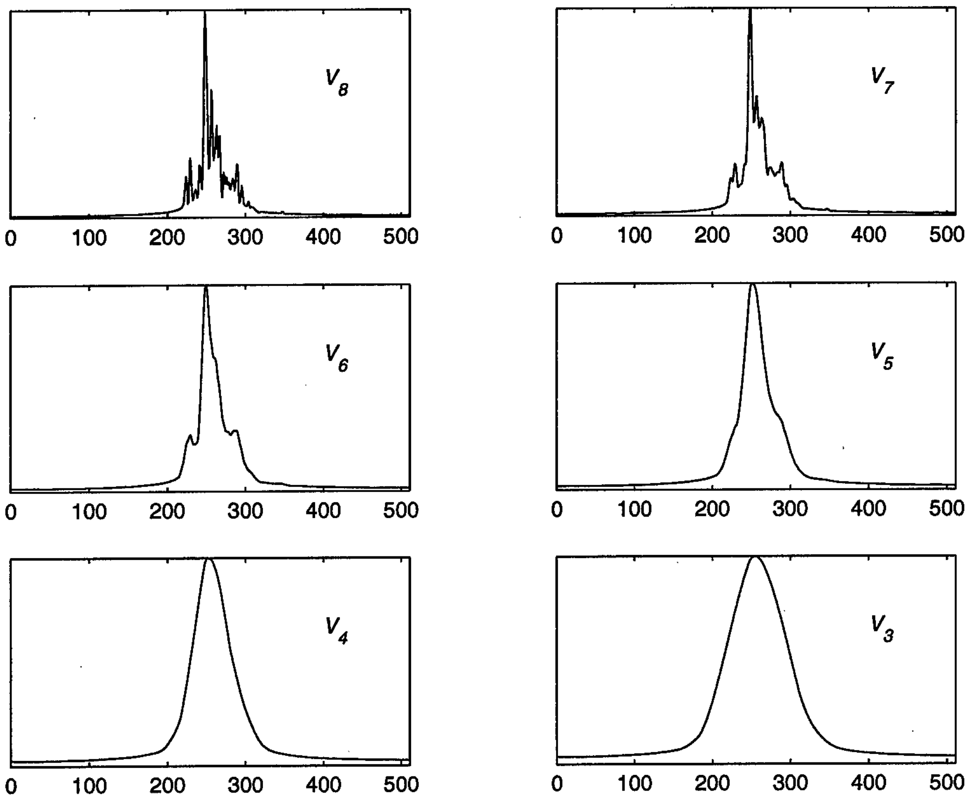


Figure 3.6 Fully translation invariant projection of synthetic HRR signature onto scaling function spaces.

particular **c** such that the threshold specifies a percentage of the coefficient in **c** whose magnitude is the largest. Setting $t = 0$ results in all coefficients remaining unchanged, whereas $t = 1$ discards all coefficients. We define $\max(\mathbf{c})$ to be the operator that returns the maximum absolute value in **c**. In effect, $\eta_t(\mathbf{c})$ applies a threshold equal to $t \max(\mathbf{c})$. Thresholding in this manner gives us the adaptability we seek. The same threshold used on two signals with one having a greater number of "large" coefficients results in a greater degree of information loss in the signal with fewer large coefficients. Note that this procedure differs from adapting a threshold to a signal, which is how adaptive wavelet thresholding is typically performed.

In Chapter 2 we mentioned that wavelet-based denoising methods typically retain all approximation coefficients. If we perform denoising in this way, then we are severely restricted. Instead, let us diverge from these methods and threshold the approximation coefficients. We specify a separate threshold for the approximation and detail coefficients and call these thresholds t_a and t_d. We thus decide to threshold the TI table in the following way:

1. For every detail column of the TI table apply the thresholding operator $\eta_{t_d}()$ as described above.
2. Apply $\eta_{t_a}()$ to the column of approximation coefficients.
3. Reconstruct signal as described in Chapter 2.

In Chapter 2 we expressed a function as

$$f(n) = \bar{R}(\mathbf{a}_{j_0}) + \sum_j \bar{R}(\mathbf{d}_j). \tag{3.6}$$

We can use a similar expression to represent our denoised signal as

$$\hat{f}(n) = \bar{R}(\eta_{t_a}(\mathbf{a}_{j_0})) + \sum_j \bar{R}(\eta_{t_d}(\mathbf{d}_j)). \tag{3.7}$$

Representing signals in this way is solely for informative purposes since in practice wavelet reconstructions are based on Mallat's algorithm. This representation gives us a qualitative feel for how we view our denoised signals: The denoised signals are viewed as the summation of a thresholded approximation portion and detail portions at successive wavelet scales. We certainly do not want to perform a full decomposition because then this representation would be meaningless since the approximation portion would be nothing more than a DC component in essence. So this representation is in line with our philosophy above regarding the choice of coarsest scales.

The above method of thresholding the TI Table is not the only reasonable method. There are numerous other possibilities. For instance, we could compute the maximum absolute value across all detail coefficients and then select t_d relative to that value (as opposed to the maximum value for each column separately). Another option would be to apply separate detail thresholds for each column, however, this would cause the complexity of the optimization procedure to grow exponentially with the number of decomposition levels. We feel that our method balances complexity and simplicity and such a balance is desirable.

3.3.5 Threshold Selection. In Chapter 2 the established techniques mentioned for selecting a threshold were based on an assumed noise model. In particular, Donoho's VisuShrink method (23) has been used extensively with success. We also mentioned that the well-established techniques set out to minimize a risk (as in Equation (2.43)) and that such methods are unsuitable in our case. We find then that we have little guidance in determining the thresholds. It certainly would not be reasonable to allow too large a value for t_a, since this would result in a near loss of the approximation. A large value for t_a would only make sense if our approximation scale was extremely coarse, but (as we saw previously) our coarsest scale corresponds to V_6. For this reason, we allow t_a to be in the range $0.0 \leq t_a \leq 0.3$. Conversely, we suspect that we can discard considerable detail, and so we restrict t_d to be in the range $0.0 \leq t_d \leq 1$. By choosing the thresholds in this way, we are able to consider the special case of signal reconstruction using the approximation projection alone, which occurs for $t_a = 0$ and $t_d = 1$.

3.3.6 Optimization. From the above, we see that our denoising scheme is composed of various parameters. We define these parameters as follows:

w: choice of wavelet

η: choice of thresholding method

j_0: choice of coarse scale

t_a, t_d: choices for approximation and detail coefficient thresholds

All the above parameters can be encapsulated into a set of parameters $\mathcal{P}_D$ defined as $\mathcal{P}_D = \{w, \eta, j_0, t_a, t_d\}$. We also define $D(f; \mathcal{P}_D)$ to be the operation of denoising the signal f using the full TI scheme (i.e. the operator that returns $\hat{f}$ as in Equation (3.7)). Since we are optimizing our denoising scheme for classification accuracy, we must decide on the classification parameters.

3.3.6.1 Classification Parameters and Simplifications. Due to the large amount of data available, we have a variety of options for performing classification. We must decide which of the six targets to use, which 5 X 5 window(s), etc. To make our problem more manageable, we make several simplifications. First, we require a forced decision so that a feature vector is always assigned to one of the training classes. In a fielded system, this would not be the case - we would add additional functionality to the classifier so that there would be an unknown class. Targets for which there is no training data could then be assigned to this class. Second, we assume that we know with certainty that a HRR signature from a given 5 X 5 window did in fact come from that window. If the classification system were operating in real time, Kalman filtering techniques would be used to estimate the azimuth and elevation at which a signature was collected (19). This process is not exact, and so there is the possibility that a signal labeled as being from a certain 5 X 5 window was in fact from a neighboring 5 X 5 window. To account for this, signatures can be compared with templates in neighboring windows.

A real time system is allowed a brief moment of time to make a decision, and during this time several signatures can be collected. Thus multiple signatures (referred to from here onward as multiple "looks") may be used to make a decision. We expect to classify more accurately using multiple looks than would be the case if we were only to classify using a single look.

We need to decide which 5 X 5 window(s) to use, which of the six targets to use for training and testing, and how many signatures to allow for classification. Our classification parameters are defined (in a manner similar to our denoising parameters) as follows:

$\mathcal{W} = \{win^1_{a,e}, \ldots, win^N_{a,e}\}$: A set of 5 X 5 windows, each with a starting azimuth and elevation a and e respectively.

$\mathcal{T} = \{c_1, \ldots, c_N\}$: A training set consisting of N classes.

nl: The number of looks employed.

We now define $C_{j,k}(D(f_j; \mathcal{P}_D); \mathcal{W}, \mathcal{T}, nl)$ to be the entire process of using the parameters above to classify a denoised signal f whose true class is j and whose assigned class is k, such that $C_{j,k} = 1$ for $k = j$, and 0 for $k \neq j$. We elaborate on these parameters below:

Set #	Targets
1	A, B, C
2	D, E, F
3	A, B, C, D, E, F

Table 3.1 Target subsets of interest

5 X 5 Window Choice. Zumwalt (54) chose a particular 5 X 5 window that was data rich, and all his results were computed for this window (which corresponds to $win_{65,15}$).We choose the same window to provide a direct comparison. In addition, we are interested in assessing the performance of our denoising scheme when considering multiple 5 X 5 windows, which gives us some idea as to robustness and generality. We consider $win_{60,15}$, $win_{60,25}$, $win_{65,15}$, $win_{70,15}$,and $win_{75,15}$. These windows correspond to the shaded regions of Figure 1.2.

Training and Testing Data. There are two major categories of data - real and synthetic - and we are interested in using both. We are particularly interested in the case of synthetic data, for it is with this data that the baseline classifier has difficulty. Among the six classes, there are three subsets of interest. See Table 3.1: Set 1 consists of three easy targets, whereas set 2 consists of three hard targets. We are solely interested in set 3 since it contains a mixture of easy and hard targets and thus provides for a more difficult problem (i.e. distinguishing amongst six targets is more difficult than distinguishing amongst three targets). Denoising is optimized for this set.

As mentioned in Chapter 1, the number of signatures vary from window to window across all classes. In addition, for each window, signatures are collected on two "tracks" that correspond to separate data collection sessions. Kosir suggests using one track for training and one for testing (32). If training and testing are performed using the same track, then the classifier could output overly optimistic results, because the signatures could exhibit more similarity for a given track than would be the case between tracks. To address the above issues, we choose track 1 for training and track 2 for testing as did Zumwalt. Then, for the case of one window, we choose the number of training and testing signatures to be equal to the minimum numbers of signatures for either track across all

Case	**Description**	**# Training/Testing Signatures per Class**
1	Real Training Data, Single Window, Single Look	51/51
2	Synthetic Training Data, Single Window, Single Look	25/51

Table 3.2 Single window cases using $win_{65,15}$

targets. When we consider multiple windows, we use this same rule separately for each window.

Incorporating Multiple Looks. Since the classification system has a brief period of time to make a decision, we use this time to collect HRR signatures in sequence. According to Fukunaga (26), this is beneficial because each signal in the sequence is of the same class, and so in theory we can average these signatures and this average will more closely resemble the class template. Although this is an attractive option, Kosir (32) found that averaging the discriminants for each signature and classifying based on the class for which the average is largest yielded better results. Thus we adopt Kosir's method.

The number of looks that we incorporate depends upon the amount of time required to make a decision and on the speed with which HRR signatures can be collected and processed. Broussard (5) stipulates that ten looks are feasible and we choose ten looks based on his assertion. In order to simplify our denoising optimization we consider only one look. Results for multiple looks are computed using the denoising parameters determined using the single look.

Based on the above discussion, there are several classification cases that we are interested in optimizing over. These cases fall into two main categories - one window and multiple windows. Tables 3.2 and 3.3 summarize these cases of interest.

		# Training/Testing Signatures per Class per Window				
Case	**Description**	$win_{60,15}$	$win_{60,25}$	$win_{65,15}$	$win_{70,15}$	$win_{75,15}$
1	Real Training Data, Multiple Windows, Single Look	60/60	64/64	51/51	50/50	45/45
2	Synthetic Training Data, Multiple Windows, Single Look	25/60	25/64	25/51	25/50	25/45

Table 3.3 Multiple window cases using $win_{60,15}$, $win_{60,25}$, $win_{65,15}$, $win_{70,15}$, and $win_{75,15}$

3.3.7 Procedure. Having formally posed our problem, we now succinctly define the quantity to be maximized as

$$A = \sum_{i=1}^{NW} \frac{NT_i}{NT_{tot}} \left(\frac{1}{NC} \left(\sum_{j=1}^{NC} \frac{1}{NT_i} \sum_{n=1}^{NT_i} C_{j,k}(D(f_j^{(n)};\mathcal{P}_D);\mathcal{W}_i,\mathcal{T},nl) \right) \right) \tag{3.8}$$

where NW is the number of 5x5 windows, NT_i is the number of testing signatures used from window $\mathcal{W}_i$, and NC is the number of test classes. The $\frac{NT_i}{NT_{tot}}$ term allows for a weighted average based on the number of testing signals used from each class. Optimizing is a matter of computing A for a large number of denoising parameter combinations and choosing the set of parameters that result in maximum A. We do this for each of the cases tabulated above. The one remaining issue is the resolution of the thresholds. Thresholds must be chosen so that the optimization procedure is completed in a reasonable amount of time. We note that we have three choices for j_0, eight choices for the wavelet, two choices for thresholding method, N_{t_d} choices for detail threshold and N_{t_a} choices for approximation threshold, which give us $3 * 8 * 2(N_{t_d} N_{t_a})$ parameter combinations. We previously decided for t_a to be in the range $0.0 \leq t_a \leq 0.3$ and for t_d to be in the range $0.0 \leq t_d \leq 1$. If we set the t_a and t_d increments to be 0.015 and 0.050, respectively, then $N_{t_a} = N_{t_d} = 21$. We end up with 21168 denoising parameter combinations. For a single window, each evaluation of Equation (3.8) takes approximately one minute and with five windows it takes approximately four minutes when running in the Matlab computational environment. Having one processor cycle through all combinations for the single window case would take two weeks for completion. Nearly two months would be needed for completion of the multiple window case. We need to perform these optimizations for both the measured and synthetic data and we see that optimization has the potential to be prohibitively time consuming. However, each evaluation of Equation (3.8) is independent, and so we take advantage of parallelism achieved by the use of multiple processors. Optimization for all cases is done in a reasonable amount of time when using multiple processors. It may seem unnecessary to pose the calculation of A in a formal manner as was done above, since we are simply going to implement the computation on a computer, but formalizing our problem can be potentially beneficial in consideration of current efforts within computer science which seek to synthesize software implementations from formal, mathematical problem descriptions.

3.4 Summary

This chapter introduced a philosophy for HRR signature denoising based on an abstract idea of noise. The approach is unconventional but more powerful than the traditional treatments of HRR classification in the presence of Gaussian noise. We presented a denoising methodology and described its optimization with respect to classification accuracy. This optimization procedure is a form of exhaustive search made possible by the high-performance of today's computational resources. The optimization procedure gives us a high level of confidence in our choice of denoising parameters since such a large number of parameter combinations are considered. In the next chapter we apply the denoising scheme to the HRR classification problem.

IV. Results

4.1 Introduction

In this chapter, we compare the results of the baseline classifier with those obtained when implementing the denoising methodology outlined in the previous chapter. Optimal denoising parameters are determined based on the six class data set using a single look. These same parameters are then used for the extension to multiple looks. This is valid since the multiple look scheme averages the discriminants across all looks as opposed to averaging the signatures and computing a single discriminant. If we were averaging the signatures, then we would certainly want to optimize over the full number of looks due to the fact that the averaging process would constitute a low pass filtering operation.

We examine the cases of training on measured data and training on synthetic data incorporating a single window. Then for validation purposes we incorporate five windows. Separate optimization needs to be done for the multiple windows, since we have no reason to suspect that optimal parameters for the single window case will yield favorable results for the multiple window case. Optimizing over multiple windows will provide us with insight into the generalization capability of the denoising scheme. Results are shown only for the full six class target set, for it is this set that is the most relevant.

4.2 Single Window

As a starting point, we examine the denoising results as applied to a single $5 \ X \ 5$ window. Our method of optimization is general enough so that incorporating multiple windows is easily handled - it simply amounts to specifying those additional windows. When we optimize over multiple windows we pay the price of having additional computational complexity and so we must choose a reasonable number of windows to optimize over.

4.2.1 Training on Measured Data. In this section, all results obtained by training on measured data and incorporating a single window are presented. It is known that the baseline classifier achieves high accuracies when using measured data for testing and hence it is unreasonable to strive for significant improvement in this area. Our interest in denoising then is not to achieve classification improvement but to achieve equivalent performance with simpler signals. By “simpler” we mean any signal form that contains considerably less content than the original signal. We do not apply quantitative measures to assess this concept of simpler. Instead we assess simplicity based on visual examination. It is not difficult though to imagine how one could approach this issue in a quantitative manner. A useful approach would be to compute compression ratios, since wavelet denoising and wavelet compression are related. They both take advantage of the unconditional basis property that characterizes wavelets. Our wavelet denoising is better done in an adaptive sense, and so there is no fixed compression ratio, but we can compute average compression ratios.

4.2.1.1 Baseline Performance - Single Look. We first limit the classifier to only one look for a classification decision. In subsequent sections we remove this constraint and allow multiple looks, which is a more realistic scenario. Table 4.1 summarizes the baseline results for the full six class target set. The diagonal elements indicate how many times a particular target was correctly classified. P_C stands for the probability of correct classification. Recall, that forced decisions are made by the classifier. In an actual implementation of the classifier, there would also be a probability of declaration, since the classifier only makes decisions when it is able to do so. To determine P_C for a particular target, the diagonal element corresponding to the target is divided by the sum along the respective row. This value is then converted to a percentage. We now have a feel for what is meant by “easy” and “hard” as we compare classification accuracies for the six targets. Notice that the classifier has a tendency to confuse target D with target E, and

Actual	Assigned Class						P_C
Class	A	B	C	D	E	F	(%)
A	51	0	0	0	0	0	100.0
B	1	50	0	0	0	0	98.0
C	4	0	39	0	5	3	76.5
D	3	3	1	27	11	6	52.9
E	1	0	1	2	47	0	92.2
F	5	0	0	2	1	43	84.3
All Classes							84.0

Table 4.1 Baseline target confusion matrix for the case of a single window, single look, and measured training data

hence Table 4.1 is referred to as a *confusion matrix.* Ideally, the confusion matrix would be strictly diagonal. Now we remove the single look constraint and allow the classifier to make a decision by incorporating up to ten looks.

4.2.1.2 Baseline Performance - Multiple Looks. From here onward, when we consider multiple looks we do not tabulate results with the detail shown in Table 4.1 except in the case of one and ten looks. Otherwise there would be an overbearing amount of data for the reader to examine which would detract from the underlying importance of the overall results. Instead, we examine the class accuracies as they evolve as a function of the number of looks. Figure 4.1 depicts this evolution graphically. We see that target D proves to be troublesome and that five looks results in maximum performance. However, in a real time system, we would have no way of knowing that five looks would be optimal for a classification decision, and in the absence of that knowledge we must use all ten looks to make the decision, since theoretically, we should expect performance to be maximal when the greatest possible number of looks are used. As with the one look case, we display the confusion matrix in Table 4.2, as a means to gain insight into the poor classification

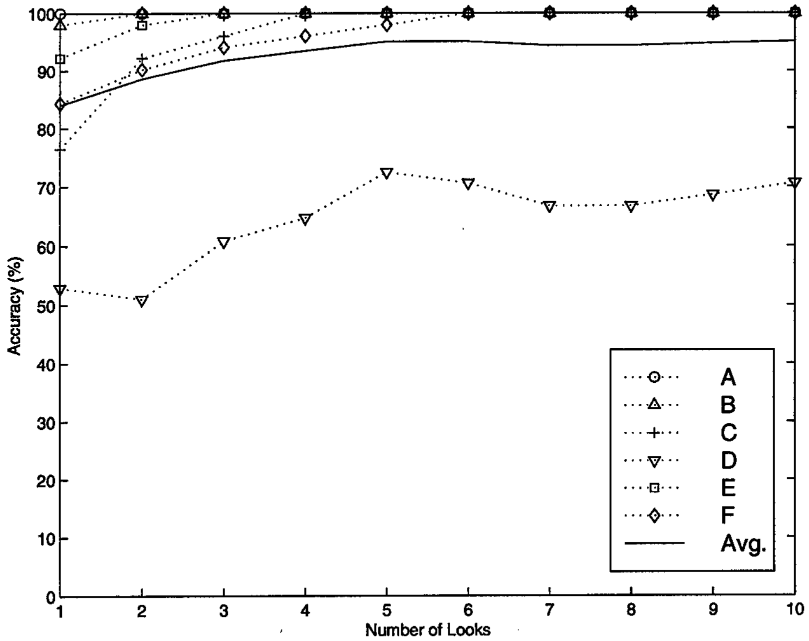


Figure 4.1 Baseline classification accuracies versus number of looks for the case of a single window and measured training data.

performance for target D. We see that the misclassifications of target D can all be attributed to confusion with target E. This confusion can be visualized by making a scatter plot as shown in Figure 4.2. Such plots show pairwise distance measures for two sets of test signatures from two classes. The line corresponds to points where the distances to the templates are equal. If we had a two class problem, then this line would be precisely the decision boundary that Equation (2.6) would dictate. When there are more than two classes, these plots serve only to show us pairwise confusion, but are nonetheless a valuable visualization tool. Ideally, the clusters would be tightly compacted and would occupy the upper left and lower right corners, signifying maximal interclass separation and minimal intraclass separation. We see from the plot that post processing of the distance measures can have the advantage of reducing misclassifications by making use of alternative decision boundaries. This issue is addressed in (26). Dewall is currently pursuing the placement of hyper-ellipses in D dimensional space, where D is the number of classes. (19).

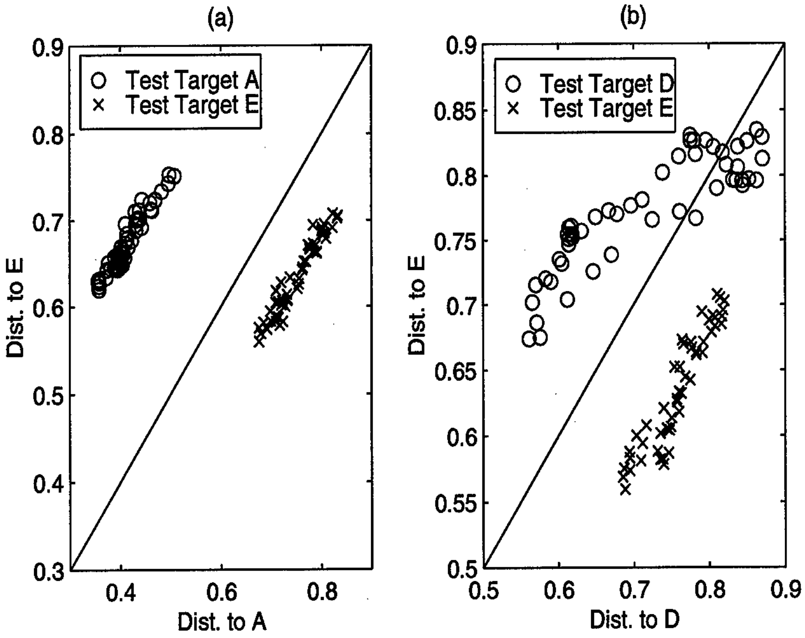


Figure 4.2 Illustration of target separability and inseparability: (a) Complete separability of targets A and E; (b) Inseparability leading to misclassifications of target D

In the following section we examine the results of the denoising optimization and repeat the classification results as above.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	51	0	0	0	0	0	100.0
B	0	51	0	0	0	0	100.0
C	0	0	51	0	0	0	100.0
D	0	0	0	36	15	0	70.6
E	0	0	0	0	51	0	100.0
F	0	0	0	0	0	51	100.0
All Classes							95.1

Table 4.2 Baseline target confusion matrix for the case of a single window, ten looks, and measured training data.

4.2.1.3 Denoising Performance - Single Look. We saw in Chapter 3 that the denoising scheme is a function of several parameters. Visualizing the classification accuracy as a function of these parameters would be a paramount feat even for the most astute topologist and is furthermore complicated since some of the parameters are categorical variables such as the wavelet choice and the thresholding method. What we can do is assume that the approximation and detail thresholds are the most significant parameters considering that these parameters can on the one extreme result in complete reconstruction and on the other extreme result in complete annihilation of a signal. The wavelet selection, for instance, could not possibly result in as widely varied a reconstruction. So for a given (t_a, t_d) pair there is a corresponding wavelet and threshold method that lead to maximum classification accuracy. Then for each decomposition level, we can plot an accuracy surface as a function of the thresholds in which case the parameterization of the wavelet and threshold method have been encapsulated as previously mentioned. Figure 4.3 shows these accuracy surfaces, where the plane surface represents the baseline accuracy of 84% using a single look.

The visualization afforded by the accuracy surface provides us with valuable insight into the denoising scheme. We see that the approximation threshold in each case has a greater effect on classification than does the detail threshold, which agrees with our intuition of the approximation coefficients providing the overall signal structure. Also in accordance with our intuition is the fact that the approximation threshold needs to be considerably smaller than that of the detail coefficients. What is surprising though is that

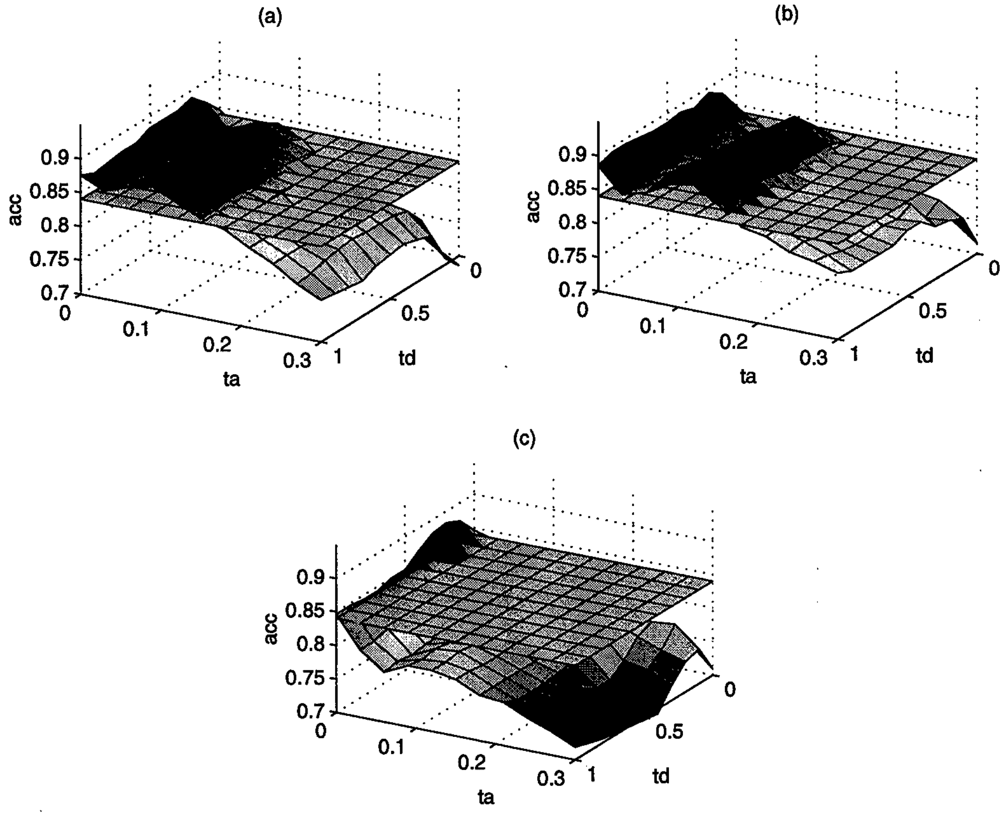


Figure 4.3 Visualization of maximum classification accuracies as a function of threshold pairs for the case of a single window and measured training data. (a) Decomposition level 1; (b) Decomposition level 2; (c) Decomposition level 3

if we consider the accuracy as a function of t_a alone, then the accuracy appears to mimic a cubic function of t_a to some extent (i.e. if we consider a slice through the $t_a - acc$ plane).

We see from Figure 4.3 that highest performance is achieved in some cases for $t_a = 0$. We can achieve nearly maximum performance for non-zero approximation thresholds and this is desirable for the following reason: Consider the accuracies in Figure 4.3(b). A maximum accuracy of 89.2% is achieved with $daub_{12}$, soft thresholding, $t_a = 0$ and $t_d = 0.3$. Now let us restrict t_a to be in the range $t_a \geq 0.05$ and we find that for $daub_{12}$, soft thresholding, $t_a = 0.12$ and $t_d = 1$ that we achieve an accuracy of 88.6%. Figure 4.4 shows a raw HRR signature along with denoised signature using the optimal parameters and the slightly suboptimal parameters. We see that what we sacrifice in classification accuracy we gain in signal simplicity.

A natural question to ask is whether or not we should also restrict t_d. We answer this in the affirmative and can provide a reasoning that serves as an alternative to the above explanation regarding the restriction on t_a. First recall that in Chapter 3 we represented a denoised signal as

$$\hat{f}(n) = R(\eta_{t_a}(\mathbf{a}_{j0})) + \sum_j R(\eta_{t_d}(\mathbf{d}_j)). \tag{4.1}$$

If we do not place a restriction on the thresholds then we would be able to achieve virtually identical performance at all decomposition levels. (If we specified separate detail thresholds then we could achieve identical performance at all levels.) To see why, we need to consider the multiresolution framework of a wavelet system, in which case for some choice of coarsest scale we have that

$$L^2 = V_{j0} \oplus W_{j0} \oplus W_{j0+1} \oplus \cdots. \tag{4.2}$$

If we choose a coarser starting scale, then we have the equivalent representation

$$L^2 = V_{j0-1} \oplus W_{j0-1} \oplus W_{j0} \oplus W_{j0+1} \oplus \cdots, \tag{4.3}$$

and so to go to a representation at a coarser scale, the approximation in V_{j0} loses some of its detail which then goes into W_{j0-1}. If we achieved a certain level of performance at one scale and needed to maintain most of the approximation, then when we go to the coarser scale, to regain the information that was contained in that approximation we would need to keep most of the information in W_{j0-1} and hence use a small t_d. Figure 4.3(c) shows this effect, as we see that to maintain the performance of the previous decomposition levels, we need to keep the approximation coefficients and most of the detail coefficients. To avoid this ambiguity in signal representation and to be consistent with the philosophy that $R(\eta_{t_a}(\mathbf{a}_{j0}))$ contains an underlying signal structure, we then restrict both t_a and t_d. This restriction also allows us to view the coarsest scale as a smoothing parameter since the restriction prevents a representation for a given coarsest scale from containing as much detail as that of the next finer scale. From here onward, accuracy surfaces are plotted for the ranges $t_a \geq 0.05$ and $t_d \geq 0.3$. It is not necessary to restrict these thresholds in the same manner since we typically can afford to lose more detail information. With the threshold restrictions placed, we can determine the denoising parameters which we still refer to as optimal since we are doing nothing more that constrained optimization. Table 4.3 contains the optimal parameters for each decomposition level. Here, the accuracy refers to the overall percentage of correct classifications (i.e., the figure in the bottom right corner of a confusion matrix). The levels correspond to the number of iterations through the filter bank implementation of the wavelet transform.

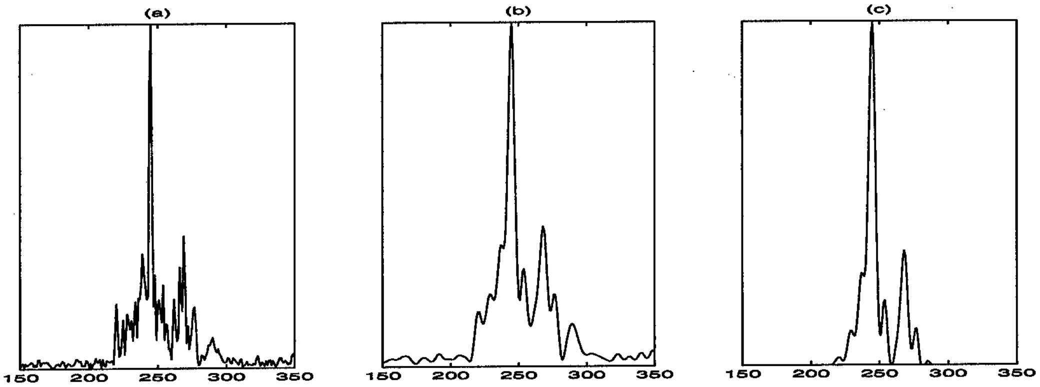


Figure 4.4 (a) HRR signature; (b) Optimally Denoised signature; (c) Sub-optimally denoised signature

We are interested to see the transformation that takes place when applying these parameters to a typical HRR signature. Figure 4.5 shows the results of denoising a typical HRR signature using the parameters listed in Table 4.3. The coarser the approximation space is, the coarser the signal representation is which is the desired property as motivated above. Note also the spatial adaptability that wavelets possess. Detail is kept where needed, and otherwise considerable smoothing occurs. Fourier domain filtering could not possibly result in such representations since detail would be kept globally, or smoothing would occur globally.

Target accuracies obtained when denoising with the three sets of parameters are shown in Table 4.4. At a first glance it would seem that denoising with one decomposition level is preferred. However, the multiple look performance needs to be evaluated to get the full picture.

Level	Accuracy (%)	Wavelet	Threshold Method	t_a	t_d
1	89.9	$daub_{16}$	Soft	0.075	0.30
2	88.6	$daub_{12}$	Soft	0.12	1.00
3	81.0	$daub_{12}$	Soft	0.09	0.35

Table 4.3 Optimal denoising parameters for the case of a single window and measured training data

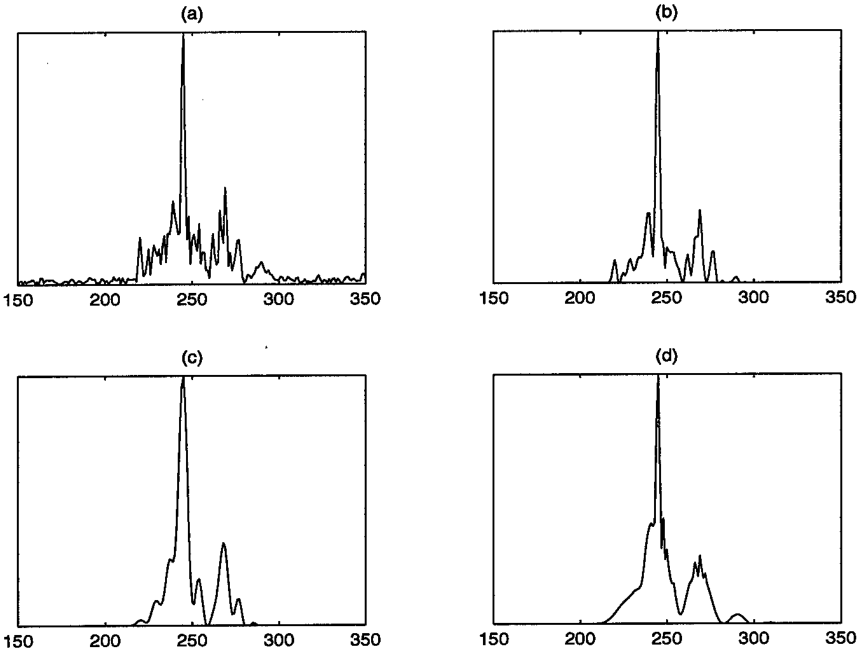


Figure 4.5 Denoised signal representations for the case of a single window and measured training data: (a) Original HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signatures using level 2 parameters; (d) Denoised signature using level 3 parameters

4.2.1.4 Denoising Performance - Multiple Looks. Classification performance is now examined as a function of the number of looks. In the previous section it was seen that single look performance is best when denoising at decomposition level one. Let us examine the ten look performances for each denoising scheme, which are shown in Table 4.5. Considering the relatively poor performance achieved when denoising at level three, it is surprising to see that this denoising scheme leads to the *best* ten look performance. Of course, it is not valid to attribute any statistical importance to an accuracy increase of 0.7%, since the number of test signatures for each target is only 51. Thus, performances in all denoising cases as are viewed as being equivalent.

	Target Accuracies (P_C)						
Level	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	100.0	100.0	92.2	56.9	98.0	92.2	89.9
2	100.0	100.0	90.2	56.9	92.2	92.2	88.6
3	98.0	98.0	84.3	56.9	82.4	66.7	81.0

Table 4.4 Target accuracies with denoising for the case of a single window, single look, and measured training data.

	Target Accuracies (P_C)						
Level	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	100.0	100.0	100.0	70.6	100.0	100.0	95.1
2	100.0	100.0	100.0	70.6	100.0	100.0	95.1
3	100.0	100.0	100.0	74.5	100.0	100.0	95.8

Table 4.5 Target accuracies with denoising for the case of a single window, ten looks, and measured training data.

For comparison purposes we show accuracies as a function of the number of looks for both the level one and the level three denoising to see the evolution that resulted in the excellent level three denoising performance. See Figure 4.6. Observe that with level one denoising, the target accuracies (with the exception of target D) quickly level off at 100%. With level three denoising, the target accuracies level off similarly, but do so only after the full ten looks. This brings up some serious philosophical issues because it must be decided which denoising scheme is ultimately preferred. Since the multiple look strategy involves averaging the discriminants from all looks, it may be reasonable to assume that, in general, the denoising scheme that works best for a single look will also work best for multiple looks, and that the results obtained above are no more than a chance occurrence. Classification using a much larger data set would need to be performed to resolve the issue with confidence. Until there is resolution to this issue we choose the coarsest level denoising scheme in the case of equivalent denoising schemes. We do so because denoising at coarser levels provides us with simpler signals which is a goal of this thesis.

The ten look confusion matrix is shown in Table 4.6. The baseline misclassifications of target D are all attributed to confusion with target E. Denoising has lessened this confusion but has added confusion with target C. The fact that the denoising process adds previously non-existent confusion suggests that there may be a benefit to optimizing denoising for

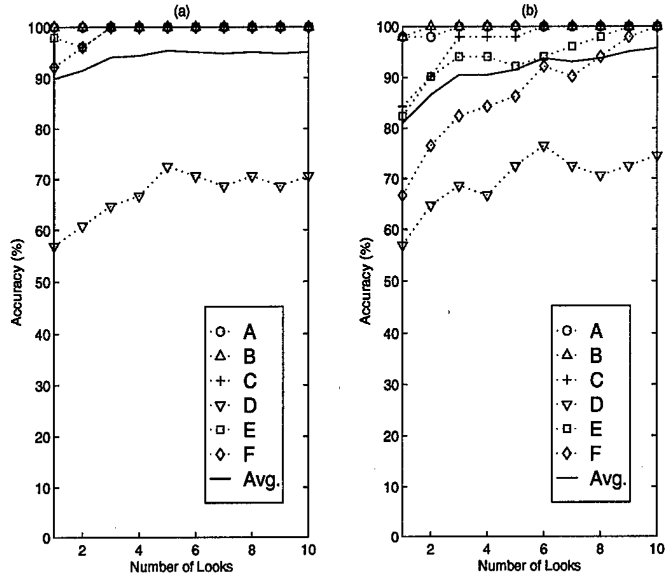


Figure 4.6 Target accuracies with denoising versus the number of looks for the case of a single window and measured training data. (a) Level one denoising performance (b) Level three denoising performance

targets individually, since targets may require different degrees of thresholding. Results are summarized in terms of improvement relative to the baseline results for the two extreme cases of one look and ten looks. See Table 4.7. The performance gains for the single look case are not of much interest. It is the ten look performance that is most relevant, and denoising performance with ten looks is essentially equivalent (at all three decomposition levels) to that of the baseline performance.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	51	0	0	0	0	0	100.0
B	0	51	0	0	0	0	100.0
C	0	0	51	0	0	0	100.0
D	0	0	6	38	7	0	74.5
E	0	0	0	0	51	0	100.0
F	0	0	0	0	0	50	100.0
All Classes							95.8

Table 4.6 Target confusion matrix with denoising for the case of a sinle window, ten looks, and measured training data.

		Targets						
	Level	A	B	C	D	E	F	Avg
1 Look	1	0.0	2.0	15.7	4.0	5.8	7.9	5.9
	2	0.0	2.0	13.7	4.0	0.0	7.9	4.6
	3	-2.0	0.0	7.8	4.0	-9.8	-17.6	-3.0
10 Looks	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	3.9	0.0	0.0	0.7

Table 4.7 Relative classification improvements for the case of a single window and measured training data.

4.2.2 Training on Synthetic Data. The baseline and denoising performance are nearly identical for the case of training on measured data. The key achievement in the measured training data case is that the denoised HRR signatures can indeed be much simpler than the original. Our focus now switches to training on synthetic data for it is with this case that baseline performance is significantly degraded as compared to the case of training on measured data. The goal now is not only to perform classification with simpler signals but to achieve a significant increase in classification accuracy as well. As in the previous section, we begin by presenting the baseline results.

4.2.2.1 Baseline Performance - Single Look. Table 4.8 contains the confusion matrix for the case of interest. The degradation in performance is quite significant and is characteristic of training on synthetic data. In the case of training on measured data, target D is the poorest performer. This is certainly not the case when training on

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	40	11	0	0	0	0	78.4
B	1	50	0	0	0	0	98.0
C	9	1	19	14	0	8	37.3
D	1	10	1	34	0	5	66.7
E	18	4	0	23	3	3	5.9
F	2	16	0	4	0	29	56.9
All Classes							57.2

Table 4.8 Baseline target confusion matrix for the case of a single window, single look, and synthetic training data.

synthetic data, as it is with target E that the classifier has the most difficult time. Target E is confused to a large extent with targets A and D. Target C also suffers severe degradation. It too has a large portion of misclassifications due to confusion with target D. We are now interested in incorporating multiple looks as a means to improve performance.

4.2.2.2 Baseline Performance - Multiple Looks. Figure 4.7 shows the multiple look performance. The confusion matrix for the full ten look classification is shown in Table 4.9. There is a great deal of disparity amongst the target accuracies. On the one extreme, targets B and D reach 100% accuracy, and on the other, target E has been entirely misclassified due to confusion with targets D and A. A primary goal of denoising then is to assuage this confusion.

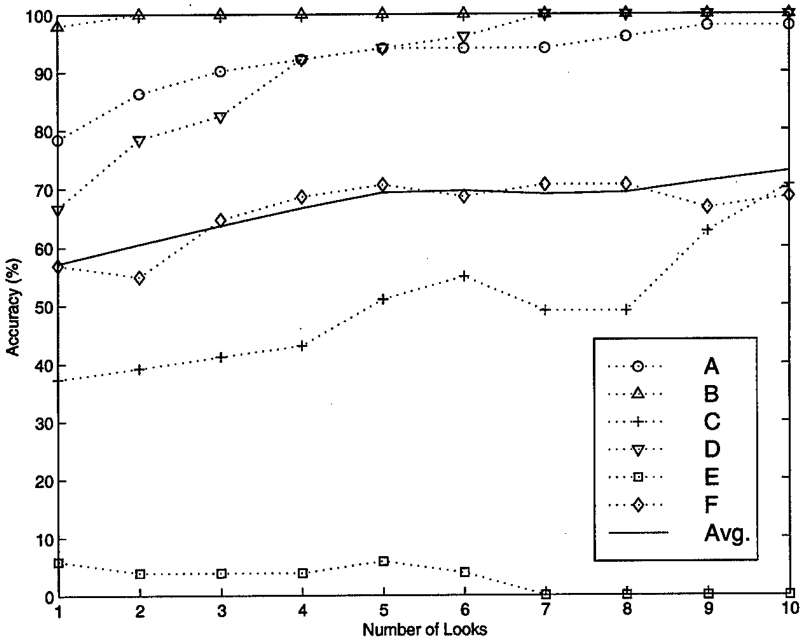


Figure 4.7 Baseline target accuracies versus number of looks for the case of a single window and synthetic training data.

4.2.2.3 Denoising Performance - Single Look. Optimal denoising parameters for the case of training on synthetic data are determined in an identical manner to the measured training data case, by restricting the thresholds as previously mentioned. The accuracy surfaces are shown in Figure 4.8. Immediately it is seen that the denoising

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	50	1	0	0	0	0	98.0
B	0	51	0	0	0	0	100.0
C	7	0	36	8	0	0	70.6
D	0	0	0	51	0	0	100.0
E	16	0	0	35	0	0	0.0
F	0	16	0	0	0	35	68.6
All Classes							72.9

Table 4.9 Baseline target confusion matrix for the case of a single window, ten looks, and synthetic training data.

scheme leads to a *significant* increase in classification accuracy. Indeed for a wide range of thresholds near optimal results are achieved. These surfaces provide some evidence of robustness since slight changes in the thresholds do not lead to a large change in accuracy. Also notice that the accuracy is in essence unaffected by the detail threshold. It may be valid then to dismiss the detail coefficients all together. However, more careful analysis would need to be done to validate such a claim and so we select the thresholds that occur at the surface maxima. The optimal denoising parameters are found to be those in Table 4.10. We see how a typical synthetic signature is transformed through the denoising process in Figure 4.9. Using the optimal parameters and a single look, we get the classification accuracies shown in Table 4.11. We now see how powerful the denoising scheme can be as we are able to improve the average baseline accuracy from 57.2% to 79.1%. The true test is whether or not significant gain is made with multiple looks.

Level	**Accuracy (%)**	**Wavelet**	**Threshold Method**	t_a	t_d
1	79.1	$daub_6$	Soft	0.135	0.80
2	75.8	$daub_8$	Soft	0.195	0.70
3	67.3	$daub_{16}$	Soft	0.165	0.90

Table 4.10 Optimal denoising parameters for the case of a single window and synthetic training data.

4.2.2.4 Denoising Performance - Multiple Looks. We proceed here as we did for the case of training on measured data. First, we show the accuracies obtained for each denoising scheme when incorporating ten looks. See Table 4.12. We see that the

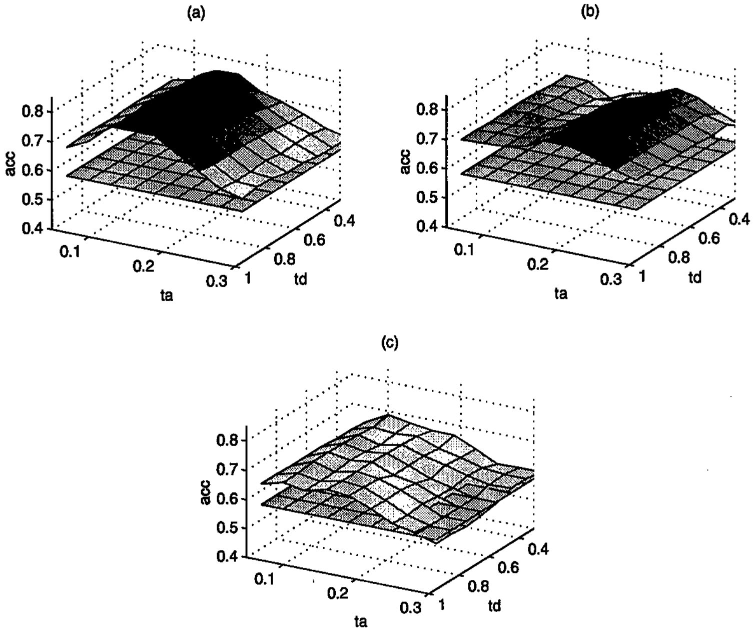


Figure 4.8 Visualization of overall target accuracies as a function of threshold pairs for the case of a single window and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2; (c) Decomposition level 3

	Target Accuracies (P_C)						
Level	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	90.0	100.0	82.4	52.9	68.6	80.4	79.1
2	90.2	98.0	74.5	47.1	62.7	82.4	75.8
3	62.7	100.0	60.8	62.7	56.9	60.8	67.3

Table 4.11 Target accuracies with denoising for the case of a single window, single look, and synthetic training data.

multiple look performance now adheres to the intuition that best single look performance should in general lead to best multiple look performance. Denoising at level one is the preferred choice since it results in a considerable higher accuracy for target D than did denoising at level 2. This is unfortunate, since we desire as coarse a signal representation as possible. The classification performance as a function of the number of looks using this scheme is shown in Figure 4.10.

Let us now display the confusion matrix that arises when using ten looks. In Table 4.13, what strikes us most is the significant improvement in classifying target E, as well as considerable improvement for targets C and E. If we compare this confusion matrix with

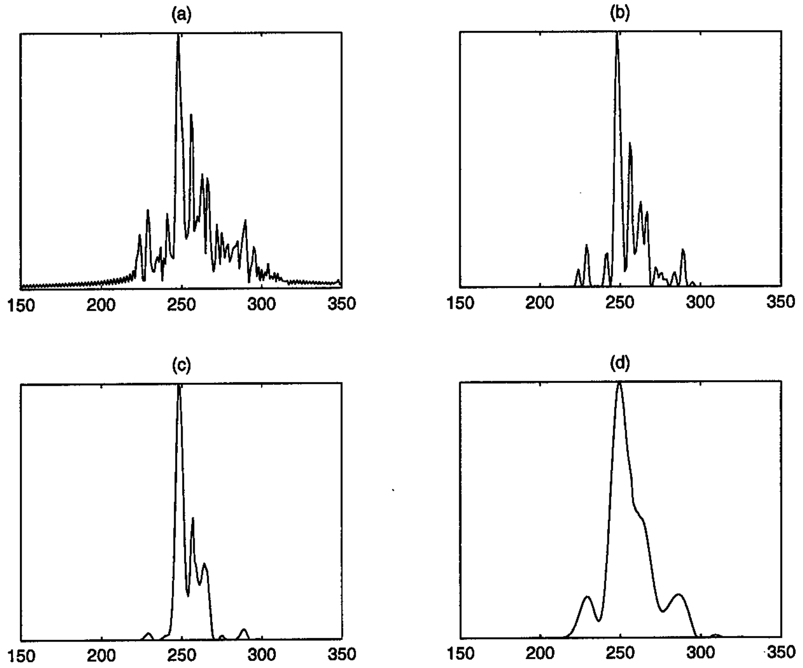


Figure 4.9 Denoised signal representations for the case of a single window and synthetic training data: (a) Original synthetic HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signatures using level 2 parameters; (d) Denoised signature using level 3 parameters

that of Table 4.2, we see that we have been able to achieve multiple look performance with synthetic training data that nearly matches that of the measured data multiple look performance. Such a result is quite surprising and is highly encouraging.

The results are not solely on the positive side as we see that there has been a degradation in performance for target D relative to the baseline result. The drastic improvement of approximately 22% in average accuracy overshadows this one negative result however. We can now gain some valuable insight through scatter plot visualization. We show the

	Target Accuracies (P_C)						
Level	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	100.0	100.0	100.0	84.3	84.3	100.0	94.8
2	100.0	100.0	100.0	74.5	84.3	100.0	93.1
3	80.4	100.0	100.0	88.2	82.4	72.5	87.3

Table 4.12 Target accuracies with denoising for the case of a single window, ten looks, and synthetic training data.

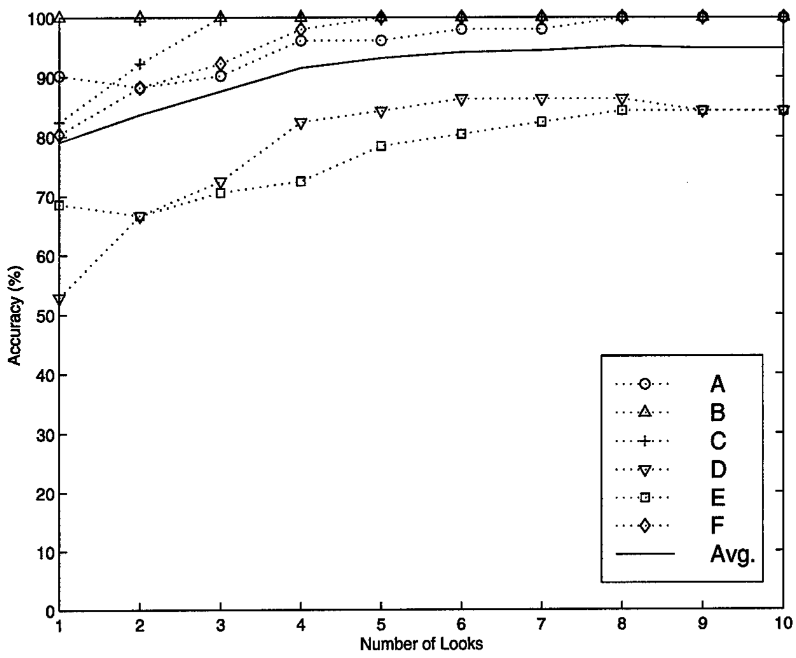


Figure 4.10 Denoising performance versus the number of looks for the case of a single window and synthetic training data.

baseline plots as well so that we have a before and after picture. Of particular interest are the scatter plots for targets A and E, D and E, and B and F, for it was with these target pairs that there was considerable baseline classifier confusion. See Figure 4.11. In the baseline case we see that target E is always "closer" to targets D and A. The corresponding denoising scatter plots reveal several important characteristics and reveal the mechanism responsible for the improved performance made possible with denoising. Denoising spreads

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	51	0	0	0	0	0	100.0
B	0	51	0	0	0	0	100.0
C	0	0	51	0	0	0	100.0
D	0	0	0	43	8	0	84.3
E	0	0	8	0	43	0	84.3
F	0	0	0	0	0	51	100.0
All Classes							94.8

Table 4.13 Target confusion matrix with denoising for the case of a single window, ten looks, and synthetic training data.

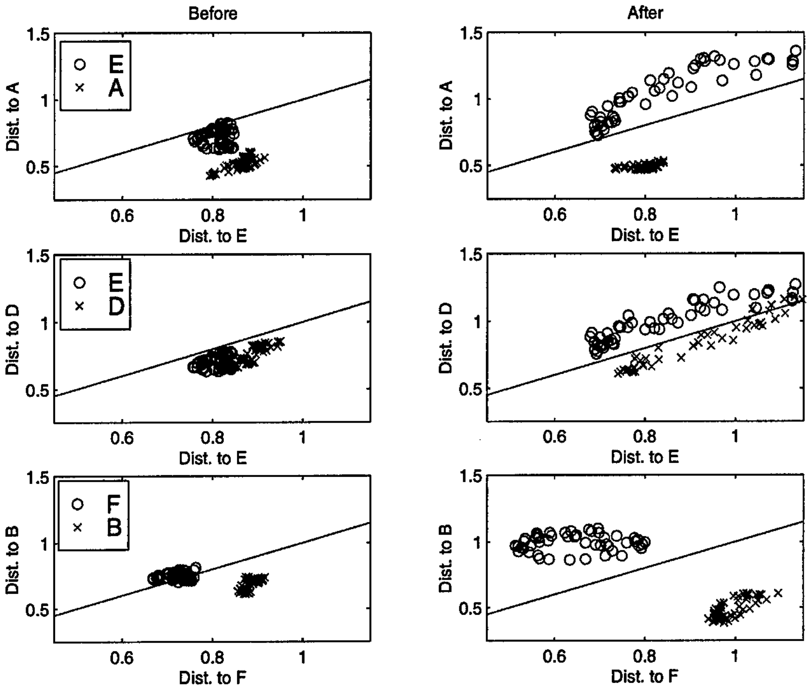


Figure 4.11 Scatter plots before and after denoising for the case of a single window, ten looks, and synthetic training data.

apart the discriminant clusters thereby increasing interclass separation, which then leads to classification improvement. However, we also see that the standard deviation of the intraclass separations has increased which is not desirable. We can correct this problem in the following manner: First, we examine a comparison of original and denoised templates for targets A and E as well as corresponding test signatures. See Figure 4.12. We see how the denoising process facilitates template comparison by making the test signatures resemble the templates more closely. However, with target E, the denoising has a tendency to leave residual peaks in the denoised test signatures as can be seen in the lower right plot in Figure 4.12. The presence of these peaks leads to an increase in the distance of the test signature to the template, which causes the intraclass distances to deviate greatly as seen in Figure 4.11. These deviations do not affect classification because the distances to the targets with which there was confusion are sufficiently large. We still desire to alleviate this problem. As previously mentioned, Dewall is applying a post-processing technique that amounts to enclosing target clusters with hyper-ellipses, and this technique is successful when clusters are compact. We must modify the denoising methodology such that residual

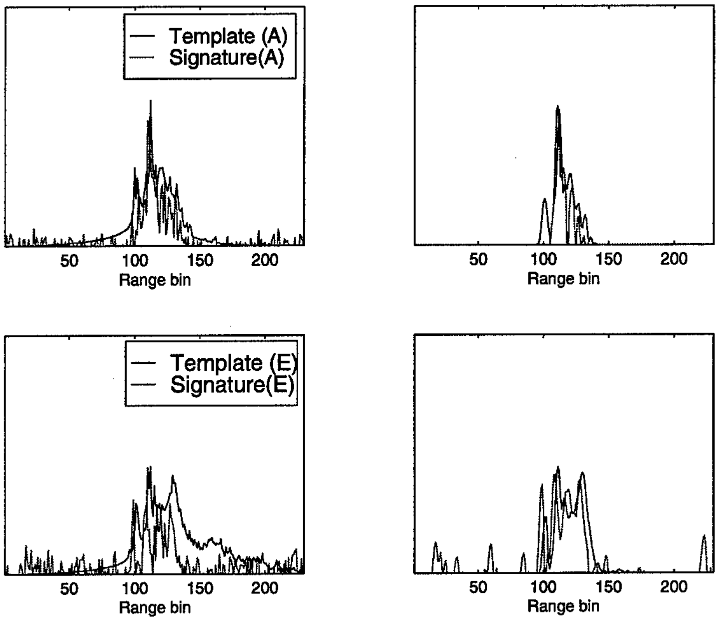


Figure 4.12 On the left, are original templates along with a test signature. On the right are templates formed by denoising along with a denoised test signature.

peaks do not appear. Though we do not implement such a modification in this thesis, we can automatically remove peaks that appear within a certain signature extent as a means to mimic the aforementioned denoising modification. If we do this and re-compute classification results, we find that accuracies remain unchanged, but we get far more desirable scatter plots as shown in Figure 4.13. Note that in general, target clusters after denoising are more compact compared to those of Figure 4.11.

To summarize, we show the denoised target classification accuracy improvement relative to the baseline improvement for the case of one and ten looks in Table 4.14. The improvement in accuracies for targets C and E are most noteworthy, but attention is also drawn to the indicated degradation in performance for target D. Target D degradation was also observed for the case of measured data. As suggested earlier, a means to prevent this degradation is to adjust the denoising parameters for targets individually. Such an endeavor is beyond the scope of this thesis, but nonetheless should be considered in future work.

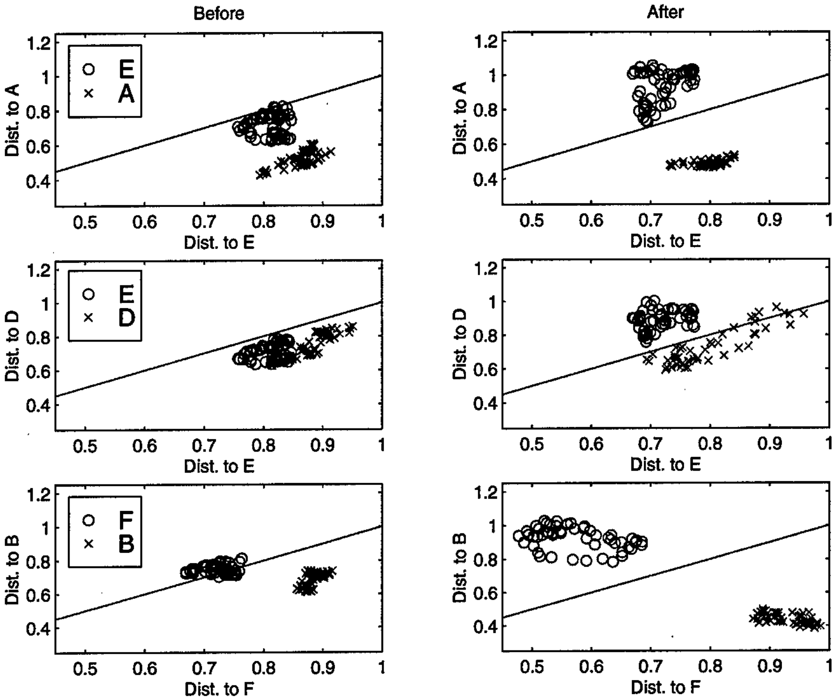


Figure 4.13 Scatter plots before and after denoising for the case of a single window, ten looks, and synthetic training data.

4.3 Multiple Windows

We have seen how powerful the denoising method of this thesis can be when applied to a single 5 X 5 window. The next logical step is to demonstrate the utility of this method when incorporating multiple windows, for it is the multiple window case that is the most relevant in the context of a real time HRR classification system. In this section we present results in a manner that is, for the most part, consistent with the presentation of results for the single window case. However, when we consider individual target accuracies, we compute averages across the windows. Similarly, when we consider the average accuracy across all targets, we compute this as an average over the five windows - that is, we compute an average of averages. Confusion matrices differ from the single window case in that the matrices are summed across all windows. These are referred to here onward as *cumulative* confusion matrices.

When we incorporated a single window, we chose the number of training and testing samples to be equal and so computing an average accuracy across all targets was straight-

		Target Accuracies (P_C)						
	Level	A	B	C	D	E	F	Avg
1 Look	1	11.8	2.0	45.1	-13.8	62.7	23.5	21.9
	2	11.8	0.0	37.2	-19.6	56.8	25.5	18.6
	3	-15.7	2.0	23.5	-4.0	51.0	3.9	10.1
10 Looks	1	14.8	0.0	45.9	-12.7	73.9	16.3	23.1
	2	14.8	0.0	45.9	-22.5	73.9	16.3	21.4
	3	-4.8	0.0	45.9	-8.8	72.0	-11.2	15.6

Table 4.14 Relative classification improvements for the case of a single window and synthetic training data.

forward. With multiple windows, the training and testing numbers are the same within each window, but these numbers vary across the windows. To compute, for instance, the average accuracy for target A across all windows, a weighted average needs to be used, in which case the weights are $N_{A,i}/N_A$, where $N_{A,i}$ is the number of target A test signatures for window i, and N_A is the total number of target A signatures across all windows. Similarly, the average overall accuracy is computed as a weighted average in which case the weights are N_i/N_{tot}. N_i is the total number of testing signatures for window i and N_{tot} is the total number of testing signatures used across all windows. Note that when computing these averages using a cumulative confusion matrix (i.e., in the same manner that is used to compute probabilities of correct classification for the single window case), the weighting occurs automatically. Computing averages in this way adheres to the NCTI Performance Reporting Standards (19).

4.3.1 Training on Measured Data. As was the case with single window classification, the intent here is not to achieve significant classification improvement, for that would be a futile goal in light of the excellent baseline results that are characteristic of training on measured data. The desire is merely to obtain equivalent results with simpler signals as was the case when incorporating a single window.

4.3.1.1 Baseline Performance - Single Look. Table 4.15 contains the single look classification accuracies for all targets across all five windows. We see that in general, performance is excellent with the exceptions of targets D and E.

	Target Accuracies (P_C)						
Window	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
$win_{60,15}$	100	98.3	81.7	66.7	66.7	98.3	85.3
$win_{60,25}$	100	90.6	85.9	79.7	62.5	95.3	85.7
$win_{65,15}$	100	98.0	76.5	52.9	92.2	84.3	84.0
$win_{70,15}$	100	100	86.0	92.0	80.0	90.0	91.3
$win_{75,15}$	100	100	86.7	62.2	60.0	91.1	83.3
Avg.							85.9

Table 4.15 Baseline target accuracies for the case of five windows, a single look, and measured training data.

To gain insight into target confusion, the cumulative confusion matrix is shown in Table 4.16. Note that there is not a particular target that is causing a large portion of the confusion, but rather the confusion is somewhat uniform. We now examine baseline multiple look performance.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	270	0	0	0	0	0	100.0
B	6	262	0	0	0	2	97.0
C	14	3	225	11	13	4	83.3
D	17	9	12	192	16	24	71.1
E	5	2	30	26	194	13	71.9
F	7	0	3	9	2	249	92.2
All Classes							85.9

Table 4.16 Cummulative baseline target confusion matrix for the case of five windows, a single look, and measured training data.

4.3.1.2 Baseline Performance - Multiple Looks. From Figure 4.14, a significant improvement is seen in the classification accuracy of the hard targets D and E resulting in an average target accuracy of approximately 97%. The individual ten look target accuracies for all windows and the cumulative confusion matrix are shown in Tables 4.17 and 4.18 respectively. The ten look results show that the confusion is almost exclusively due to an equal amount of confusion between targets D and E. Recall from the single window results that target E was confused to a large extent with target D but not the other way around. We assume that the multiple look results are more indicative of a general trend and that targets D and E both tend to experience a similar degree of confusion.

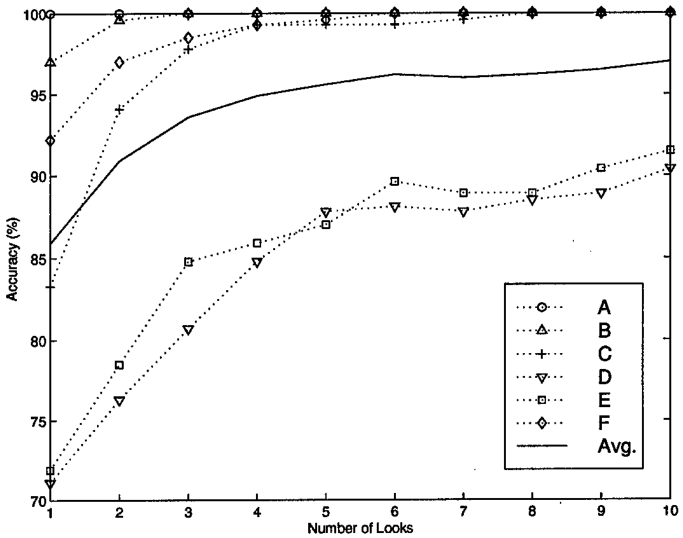


Figure 4.14 Average baseline target accuracies versus number of looks for the case of five windows and measured training data.

	Target Accuracies (P_C)						
Window	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
$win_{60,15}$	100	100	100	98.3	88.3	100	97.8
$win_{60,25}$	100	100	100	93.8	85.9	100	96.6
$win_{65,15}$	100	100	100	70.6	100	100	95.1
$win_{70,15}$	100	100	100	100	100	100	100
$win_{75,15}$	100	100	100	86.7	84.4	100	95.2
Avg.							97.0

Table 4.17 Baseline target accuracies for the case of five windows, ten looks, and measured training data.

The denoising goal is to achieve results equivalent to those above using simpler signal representations. It is also desired to alleviate the target confusion seen in Table 4.18, but it is acknowledged that such a goal may not be reached based on the single window results.

4.3.1.3 Denoising Performance - Single Look. Classification accuracy surfaces are viewed much in the same way as was done for the single window case. These surfaces are displayed in Figure 4.15. In this case, the surfaces represent the average overall accuracy obtained across all windows. By comparing Figures 4.15 and 4.3, the same overall

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	270	0	0	0	0	0	100
B	0	270	0	0	0	0	100
C	0	0	270	0	0	0	100
D	1	0	4	244	15	6	90.4
E	0	0	8	15	247	0	91.5
F	0	0	0	0	0	270	100
All Classes							97.0

Table 4.18 Cummulative baseline target confusion matrix for the case of five windows, ten looks, and measured training data.

behavior as a function of the thresholds is seen. However, behavior for the case of multiple windows appears more regularized and does not exhibit the cubic like behavior that was seen in the single window case. It is therefore assumed that the multiple window results are more representative of a general trend. The observed trend is that the accuracy is almost exclusively a function of the approximation threshold and that it decreases monotonically with increasing t_a. This trend makes sense intuitively if we recall the representation of a signal as $\hat{f}(n) = R(\eta_{t_a}(\mathbf{a}_{j_0})) + \sum_j R(\eta_{t_d}(\mathbf{d}_j))$, where the threshold restrictions forced $R(\eta_{t_a}(\mathbf{a}_{j_0}))$ to constitute the underlying signal structure. Table 4.19 contains the optimal denoising parameters at the three decomposition levels.

Level	**Accuracy (%)**	**Wavelet**	**Threshold Method**	t_a	t_d
1	89.4	$daub_{12}$	Soft	0.075	1.00
2	88.8	$daub_6$	Soft	0.075	1.00
3	85.5	*haar*	Soft	0.135	0.30

Table 4.19 Optimal denoising parameters for the case of five windows and measured training data.

The interest now is in visually examining the appearance of a typical HRR signature when denoised using the optimal parameters. Compare Figures 4.16 and 4.5. Note that the denoised signatures in both figures are nearly identical to the eye.

The optimal parameters from Table 4.19 are applied for classification and we obtain the averaged target accuracies of Table 4.20. These results are consistent with the trend of Table 4.4, in that single look accuracy drops off as the approximations become coarser. Recall that this trend was not present for the case of a single window when we incorporated

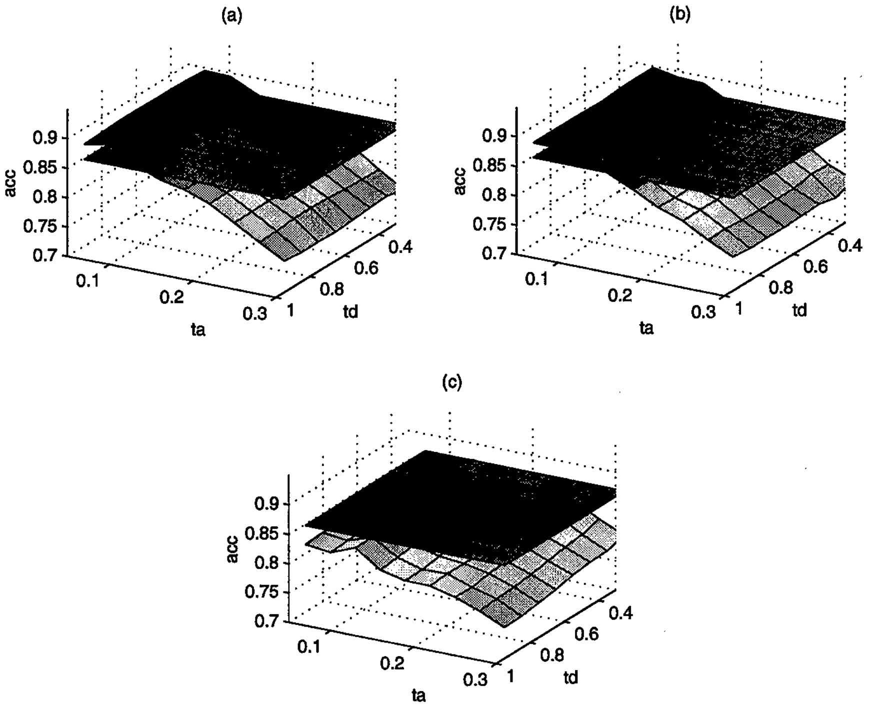


Figure 4.15 Visualization of average overall target accuracies as a function of threshold pairs for the case of multiple windows and measured training data. (a) Decomposition level 1; (b) Decomposition level 2; (c) Decomposition level 3

multiple looks, and that denoising performances at all levels were nearly equivalent and indeed counterintuitive. We now see if this trend is also present for the case of multiple windows.

	Target Accuracies (P_C)						
Level	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	98.9	98.1	89.3	72.2	82.6	95.6	89.4
2	97.4	96.7	90.4	75.9	78.1	94.4	88.8
3	91.5	94.1	83.0	69.3	83.0	91.9	85.4

Table 4.20 Average target accuracies with denoising for the case of five windows, a single look, and measured training data.

4.3.1.4 Denoising Performance - Multiple Looks. Upon examination of Table 4.31, we see denoising performances at all levels are indeed equivalent as was the case with a single window and multiple looks. Perhaps this phenomenon can be explained by recognizing that when performing multiple look classification, there is not a lot of room for

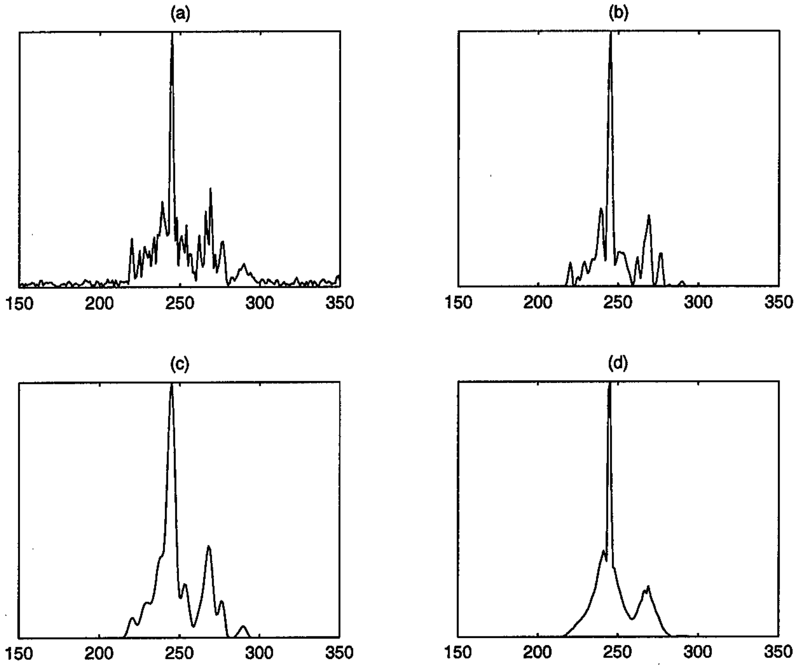


Figure 4.16 Denoised signal representations for the case of five windows and measured training data: (a) Original measured HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signatures using level 2 parameters; (d) Denoised signature using level 3 parameters

improvement when denoising at levels one and two, whereas there is relatively more room for improvement when denoising at level three. So with all else equal, the performances reach a similar steady state, much as runners in a track event can cross the finish line neck and neck even though some runners could have been strides ahead of others during the brief moment following the start of the race. The equivalent performances suggest that we should prefer denoising at level three since it allows for simpler signal forms. In Figure 4.17, we see the progression of classification using multiple looks and the level three parameters.

	Target Accuracies (P_C)						
Level	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	100	100	100	89.6	96.3	100	97.7
2	100	100	100	91.9	95.2	100	97.8
3	100	100	100	87.0	99.6	100	97.8

Table 4.21 Average target accuracies with denoising for the case of five windows, ten looks, and measured training data

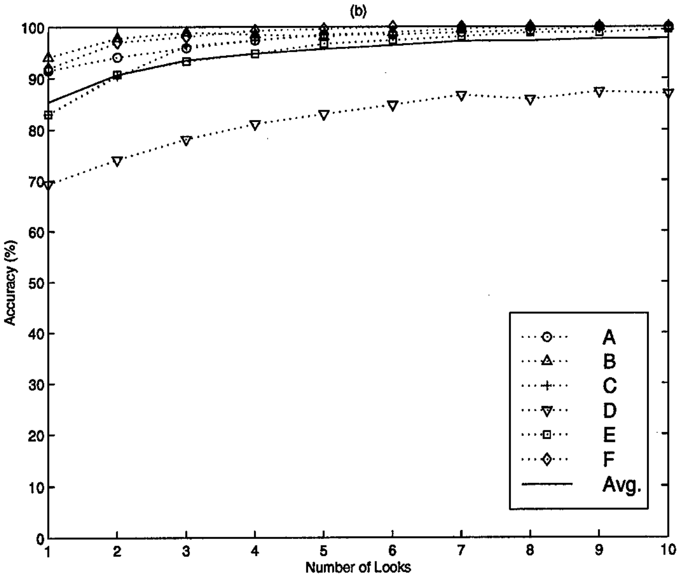


Figure 4.17 Average target accuracies with denoising versus number of looks for the case of five windows and measured training data.

Tables 4.22 and 4.23 contain the more detailed results for classification with ten looks. By comparison with Table 4.17, we see that target accuracies reach similar values, though there is a slight drop in target's D's accuracy and a corresponding increase in target E's accuracy of nearly 10%. These changes do not affect the average accuracy significantly and so it must be kept in mind that changes in target accuracies on the order of 10% can lead to changes in the average overall accuracy on the order of 1%. From the confusion matrix we also see that target E is no longer being confused with target D, though some confusion has been introduced which results a slight degradation of target D's accuracy. This effect was also observed for the case of a single window.

A comparison of multiple window performance for measured training data is in Table 4.24. Comparing the ten look performance is of the most interest and it is seen that performances are nearly identical. This is as expected, but the results are still significant since we are able to classify with simpler signal representations.

4.3.2 Training on Synthetic Data. The denoising performances obtained for the case of a single window are remarkable and the goal now is to achieve significant improvement in the case of multiple windows. Superior performance over multiple windows

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	270	0	0	0	0	0	100.0
B	0	270	0	0	0	0	100.0
C	0	0	270	0	0	0	100.0
D	0	0	13	235	10	12	87.0
E	0	0	0	1	269	0	99.6
F	0	0	0	0	0	270	100.0
All Classes							97.8

Table 4.22 Cummulative target confusion matrix with denoising for the case of five windows, ten looks, and measured training data.

	Target Accuracies (P_C)						
Window	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	100	100	100	100	98.3	100	99.7
2	100	100	100	90.6	100	100	98.4
3	100	100	100	66.7	100	100	94.4
4	100	100	100	100	100	100	100
5	100	100	100	73.3	100	100	95.6
Avg.							97.8

Table 4.23 Target accuracies with denoising for the case of five windows, ten looks, and measured training data

would suggest generalization capabilities of the denoising scheme and we now set out to demonstrate such capabilities. First the baseline performance is established.

4.3.2.1 Baseline Performance - Single Look. Table 4.25 contains the target accuracies for the various windows. The cumulative confusion matrix is shown in Table 4.26. From comparison with Table 4.8, we see that the overall multiple window performance is similar to the single window performance. In particular, we see that target E poses problems across all windows. Multiple look results are now presented.

4.3.2.2 Baseline Performance - Multiple Looks. Average target accuracies are displayed as a function of the number of looks in Figure 4.18. We see that targets B and D are the only exceptional performers. Table 4.27 contains the baseline target accuracies. Note that multiple looks result in a further degradation for target E which is contrary to what we expect from multiple look classification. A confusion matrix can add

		Targets						
	Level	A	B	C	D	E	F	Avg
1 Look	1	-1.1	1.1	6.3	1.1	10.7	3.4	3.5
	2	-2.6	-0.3	7.4	4.8	6.2	2.2	2.9
	3	-8.5	-2.9	0.0	-1.8	11.1	-0.3	-0.5
10 Looks	1	0.0	0.0	0.0	-0.8	4.8	0.0	0.7
	2	0.0	0.0	0.0	1.5	3.7	0.0	0.8
	3	0.0	0.0	0.0	-3.4	8.1	0.0	0.8

Table 4.24 Relative classification improvements for the case of five windows and measured training data.

	Target Accuracies (P_C)						
Window	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
$win_{60,15}$	63.3	95.0	36.7	75.0	1.7	88.3	60.0
$win_{60,25}$	40.6	85.9	76.6	84.4	26.6	92.2	67.7
$win_{65,15}$	78.4	98.0	37.3	66.7	5.9	56.9	57.2
$win_{70,15}$	88.0	98.0	18.0	92.0	16.0	56.0	61.3
$win_{75,15}$	95.6	100	13.3	84.4	8.9	46.7	58.1
Avg.							61.2

Table 4.25 Baseline target accuracies for the case of five windows, a single look, and synthetic training data.

to our understanding of the problem. See Table 4.28. There are six primary instances of confusion: A with B, D with C, E with A, E with D, E with F, and F with D. The misclassifications of target E are attributed to confusion between it and most other classes. It is possible that the multiple looks allow for this large degree of confusion to manifest itself fully; afterall, if you light a fire in a forest, it will spread rapidly. We return to the confusion issue in the subsequent denoising section with multiple looks.

4.3.2.3 Denoising Performance - Single Look. The accuracy surfaces that we have become accustomed to are displayed in Figure 4.19. Compare these surfaces with those of Figure 4.8, and note that the behaviors are different at all decomposition levels. This suggests that if the denoising scheme is to be implemented in a fielded system in which case a large number of windows must be incorporated, then we must maintain a large portion of the approximation coefficients. Recall, however, that the optimization process for multiple windows is designed to maximize the average overall target accuracy

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	191	52	15	12	0	0	70.7
B	7	256	0	5	0	2	94.8
C	39	8	105	92	0	26	38.9
D	8	28	10	217	0	7	80.4
E	57	9	24	103	33	44	12.2
F	19	22	4	33	2	190	70.4
All Classes							61.2

Table 4.26 Cummulative baseline target confusion matrix for the case of five windows, a single look, and synthetic training data.

	Target Accuracies (P_C)						
Window	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
$win_{60,15}$	80.0	100	61.7	91.7	0.0	100.0	72.2
$win_{60,25}$	57.8	100	96.9	100	43.8	100	83.1
$win_{65,15}$	98.0	100	70.6	100	0.0	68.6	72.9
$win_{70,15}$	100	100	14.0	100	0.0	82.0	66.0
$win_{75,15}$	100	100	8.9	93.3	0.0	57.8	60.0
Avg.							71.7

Table 4.27 Baseline target accuracies for the case of five windows, ten looks, and synthetic training data

across all windows using a single set of denoising parameters. An alternative method is addressed in a later section.

The optimal parameters are found to be those in Table 4.29. The interest now is examining the effects that denoising with these parameters has on a typical synthetic signature. These effects are seen in Figure 4.20. By comparing these denoised signatures with those in Figure 4.9, we see that we do not obtain signal realizations as simple as those in the single window case, which is due to a lesser degree of approximation coefficient thresholding.

Accuracies for the individual targets are shown in Table 4.30. We now examine the multiple look results.

4.3.2.4 Denoising Performance - Multiple Looks.

Averaged target accuracies are in Table 4.31. We see that overall accuracy drops as the signals become more coarse. This was the result we observed in the case of a single window. The performances

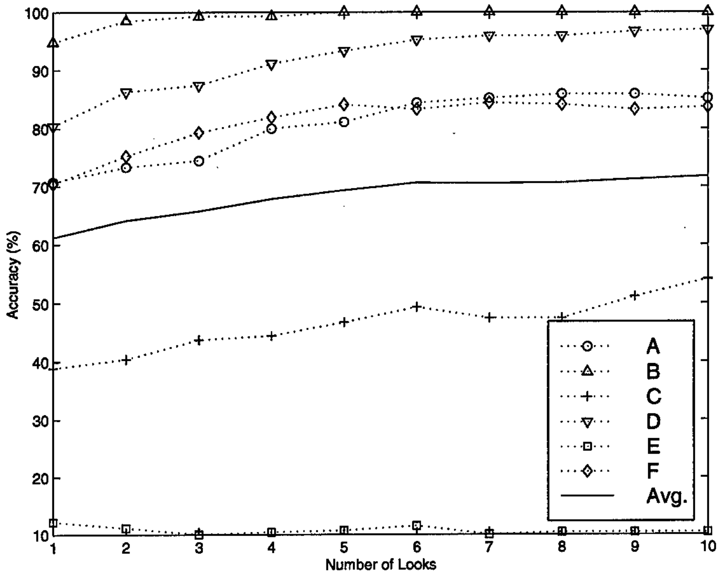


Figure 4.18 Average baseline target accuracies versus number of looks for the case of five windows and synthetic training data.

when denoising at levels one and two are for all intents equivalent, and so we adopt the level two denoising parameters for the usual reason that it affords us simpler signal representations.

The average accuracies versus the number of looks are shown in Figure 4.21. Target E proves to be the limiting factor in overall target accuracy. This is certainly more prevalent with the baseline classifier, but nonetheless, classifying target E correctly only 40% of the time is clearly not an impressive result even if it is nearly a 30% improvement over the baseline classifier. See Table 4.32 for the accuracies for each window.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	230	22	9	9	0	0	85.2
B	0	270	0	0	0	0	100.0
C	15	3	146	99	0	7	54.1
D	0	8	0	262	0	0	97.0
E	46	0	10	146	28	40	10.4
F	0	16	0	28	0	226	83.7
All Classes							71.7

Table 4.28 Cummulative baseline target confusion matrix for the case of five windows, ten looks, and synthetic training data.

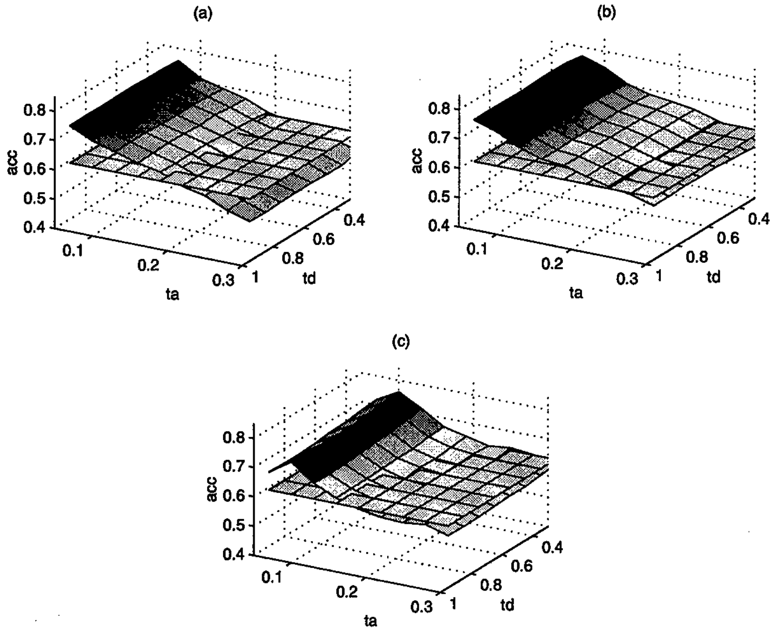


Figure 4.19 Visualization of average overall target accuracies as a function of threshold pairs for the case of multiple windows and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2; (c) Decomposition level 3

Level	**Accuracy (%)**	**Wavelet**	**Threshold Method**	t_a	t_d
1	74.0	$daub_4$	Soft	0.060	0.90
2	75.8	$daub_4$	Soft	0.060	0.40
3	74.0	*haar*	Soft	0.075	0.35

Table 4.29 Optimal denoising parameters for the case of five windows and synthetic training data.

Recall that there were six key instances of confusion for the baseline classifier. Let us examine the ten look cumulative confusion matrix that denoising produces and compare with that of Table 4.28. We see that the confusions of target E with A and target F with D have been completely removed. The confusion of target C with D has been extensively lessened and the confusions of target E with D and target E with F have been alleviated somewhat. The denoising does, however, introduce confusion of target E with C which was not present with the baseline classifier. A similar phenomenon occurred in the case of a single window and this then appears to be a general denoising result that is due to optimization for maximum averaged performance.

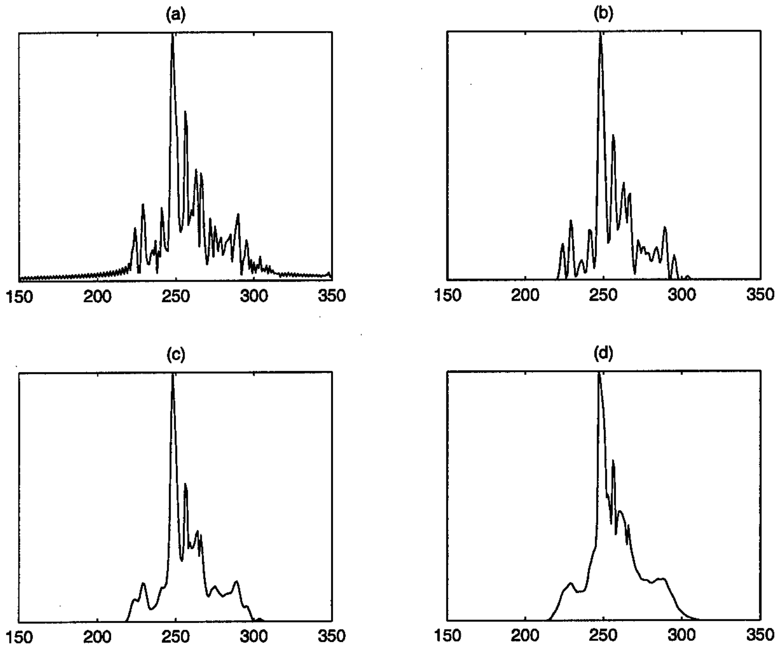


Figure 4.20 Denoised signal representations for the case of five windows and synthetic training data: (a) Original measured HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signatures using level 2 parameters; (d) Denoised signature using level 3 parameters

The relative improvements that denoising provides us are shown in Table 4.34. We see that when testing over multiple windows we achieve a remarkable level of improvement over the baseline classifier. We now have enough evidence in favor of the denoising scheme that demonstrates generalization capability. If we compare these results with those in Table 4.14, then we see a recurring trend: Multiple look accuracy degrades as the denoised signals become coarser. This is not the case when training on measured data in which case multiple look accuracies were nearly equivalent for all decomposition levels (if decompositions proceeded beyond levels 1-3, then this certainly would not be true). We can make

	Target Accuracies (P_C)						
Level	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	79.6	95.6	74.8	72.2	39.6	82.2	74.0
2	88.9	98.1	73.0	77.0	33.7	84.1	75.8
3	84.1	97.4	66.7	79.3	33.7	82.6	74.0

Table 4.30 Average target accuracies with denoising for the case of five windows, a single look, and synthetic training data.

	Target Accuracies (P_C)						
Level	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
1	94.4	100	94.8	96.3	45.2	100	88.5
2	97.8	100	92.2	95.2	40.0	100	87.5
3	97.4	100	90.4	91.1	38.9	97.8	85.9

Table 4.31 Average target accuracies with denoising for the case of five windows, ten looks, and synthetic training data

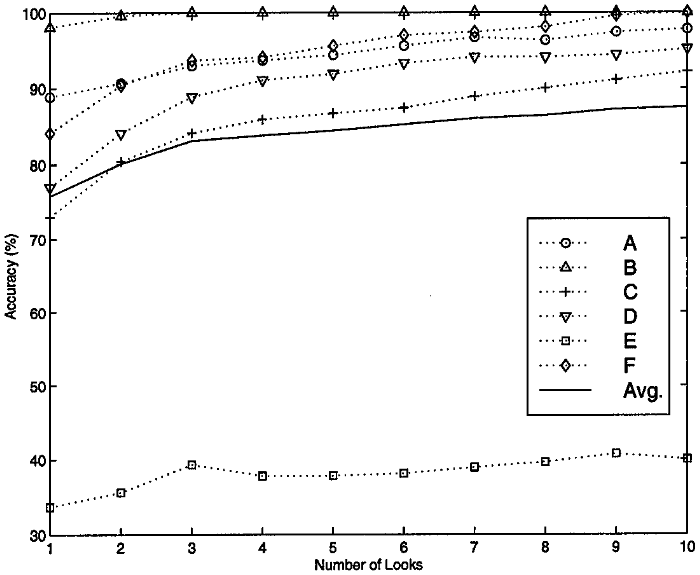


Figure 4.21 Average target accuracies with denoising versus number of looks for the case of five windows and synthetic training data.

sense of this result by considering the fact that when we train on synthetic data, we are at a disadvantage right from the beginning because we then need to match the measured signatures with these synthetically generated templates. By virtue of the modeling process, the measured signatures certainly differ to a greater extent from these synthetic templates than they would from measured templates.

4.4 Additional Considerations

In this section we address several questions that inevitably arise concerning the denoising methodology. We would like to know if the results are as impressive when using larger testing sets and larger numbers of windows. Also, we are interested to examine classification performance when implementing wavelet denoising methods that are popular

	Target Accuracies (P_C)						
Window	**A**	**B**	**C**	**D**	**E**	**F**	**Avg.**
$win_{60,15}$	100	100	98.3	100	0.0	100	83.1
$win_{60,25}$	100	100	100	87.5	93.8	100	96.9
$win_{65,15}$	88.2	100	96.1	98.0	0.0	100	80.4
$win_{70,15}$	100	100	98.0	100	34.0	100	88.7
$win_{75,15}$	100	100	62.2	91.1	68.9	100	87.0
Avg.	97.8	100	92.2	95.2	40.0	100	87.5

Table 4.32 Target accuracies with denoising for the case of five windows, ten looks, and synthetic training data

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	262	6	0	0	0	0	97.8
B	0	270	0	0	0	0	100
C	0	0	249	6	0	15	92.2
D	0	5	8	257	0	0	95.2
E	0	0	98	43	108	21	40.0
F	0	0	0	0	0	270	100.0
All Classes							87.5

Table 4.33 Cummulative target confusion matrix with denoising for the case of five windows, ten looks, and synthetic training data.

in the wavelet literature. Other questions are related more specifically to the denoising method of this thesis. For instance, how sensitive are accuracies with respect to the wavelet choice? What is gained in implementing the denoising method with translation invariance as opposed to an analogous non-translation invariant method? Lastly, we are interested in an alternative optimization method for multiple windows as was alluded to earlier. Each of these questions are now addressed.

4.4.1 Performance with Larger testing Sets. The decision to limit the number of testing signatures for each class was based mainly on a desire to lessen the computational burden. We can, however, examine some special cases as a means to put to rest any reservations that we may have concerning the performance when faced with larger test sets. Let us consider the case of training on synthetic data, for it is this case that is of the most interest.

		Target Accuracies (P_C)						
	Level	A	B	C	D	E	F	Avg
1 Look	1	8.9	0.8	35.9	-8.2	27.4	11.8	12.8
	2	18.2	3.3	34.1	-3.4	21.5	13.7	14.6
	3	13.4	2.6	27.8	-1.1	21.5	12.2	12.8
10 Looks	1	9.2	0.0	40.7	-0.7	34.8	16.3	16.8
	2	12.6	0.0	38.1	-1.8	29.6	16.3	15.8
	3	12.2	0.0	36.3	-5.9	28.5	14.1	14.2

Table 4.34 Relative classification improvements for the case of five windows and synthetic training data.

4.4.1.1 Single Window. First let us consider a single window. We have seen that results for single and multiple windows follow the same basic trends and so we have confidence in asserting that the results we obtain here carry over to the multiple window case. Confusion matrices are computed for a single look as well as ten looks. Tables 4.35 and 4.36 contain these matrices.

Actual Class	Assigned Class A	B	C	D	E	F	P_C (%)
A	235	66	0	0	0	0	78.1
B	3	308	0	0	0	0	99.0
C	22	2	43	47	0	9	35.0
D	5	32	1	192	0	17	77.7
E	18	4	0	23	3	3	5.9
F	7	42	0	27	1	75	49.3
All Classes:							57.5

Table 4.35 Baseline target confusion matrix for the case of a single window, single look, synthetic training data, and all available testing data.

Let us now examine the denoising results. See Tables 4.37 and 4.38. We see that denoising does in fact yield a remarkable improvement over the baseline results. The relative improvements are summarized in Table 4.39. These results are remarkable, however there is a drawback in that there is a degradation in performance for target D. This is a general result imposed by the optimization method and a likely solution to the problem is an optimization method that treats targets individually.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	279	22	0	0	0	0	92.7
B	0	311	0	0	0	0	100.0
C	7	0	63	53	0	0	51.2
D	0	2	0	245	0	0	99.2
E	16	0	0	35	0	0	0.0
F	3	49	0	21	0	79	52.0
Overall Accuracy:							65.8

Table 4.36 Baseline target confusion matrix for the case of a single window, ten looks, synthetic training data, and all available testing data.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	245	53	0	2	0	1	81.4
B	1	308	0	0	2	0	99.0
C	3	0	95	20	4	1	77.2
D	7	10	26	136	45	23	55.1
E	0	1	8	6	35	1	68.6
F	2	13	11	6	25	95	62.5
Overall Accuracy:							74.0

Table 4.37 Target confusion matrix with denoising for the case of a single window, single look, synthetic training data, and all available testing data.

4.4.1.2 Multiple Windows. We now incorporate multiple windows, but we do not limit ourselves to the same five windows that we considered previously. Now we make the problem more difficult. We consider a total of 12 windows (five of which are the windows previously considered), covering the azimuth and elevation span shown in Figure 4.22. and denoise using the optimal parameters found for the five window case using synthetic training data. This serves as a rigorous means to determine the robustness of the denoising method, since we now test over windows for which denoising was not optimized. As in the previous section, we show confusion matrices for one and ten look classification and compare average overall target accuracies. See Tables 4.40 and 4.41. Denoising results are in the confusion matrices in Tables 4.42 and 4.43. Again we see the trend that we are accustomed to: Target E has improved at the expense of degradation in target D, though there has been significant overall improvement. Classification with ten looks has yielded nearly a 16% improvement relative to baseline performance. This

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	289	12	0	0	0	0	96.0
B	0	311	0	0	0	0	100.0
C	0	0	118	5	0	0	95.9
D	0	0	1	217	29	0	87.9
E	0	0	8	0	43	0	84.3
F	0	0	0	0	17	135	88.8
Overall Accuracy:							92.2

Table 4.38 Target confusion matrix with denoising for the case of a single window, ten looks, synthetic training data, and all available testing data.

	Targets						
# Looks	A	B	C	D	E	F	Avg
1 Look	3.3	0.0	42.2	-22.6	62.7	13.2	16.5
10 Looks	3.3	0.0	44.7	-11.3	84.3	36.8	26.4

Table 4.39 Relative classification improvements for the case of five windows, synthetic training data, and all available testing data.

figure is particularly impressive due to the extensive testing that was done and the fact that the denoising parameters were optimized across only five windows. The results in this section lend overwhelming evidence as to the robustness of the denoising method, and also strongly support the intuition regarding simple signal representations.

4.4.2 Wavelet Sensitivity. The interest is to now assess performance sensitivity with respect to the wavelet choice. To do so, let us again consider the interesting case of training on synthetic data where we limit the scope to a single window. We apply at

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	2048	450	96	27	0	16	75.2
B	22	1529	0	9	0	3	97.8
C	115	29	440	312	3	107	43.7
D	27	195	26	1115	1	59	78.4
E	112	69	53	264	94	140	12.8
F	58	119	25	187	24	646	61.0
All Classes:							61.5

Table 4.40 Cummulative baseline target confusion matrix for the case of 12 windows, one look, and synthetic training data.

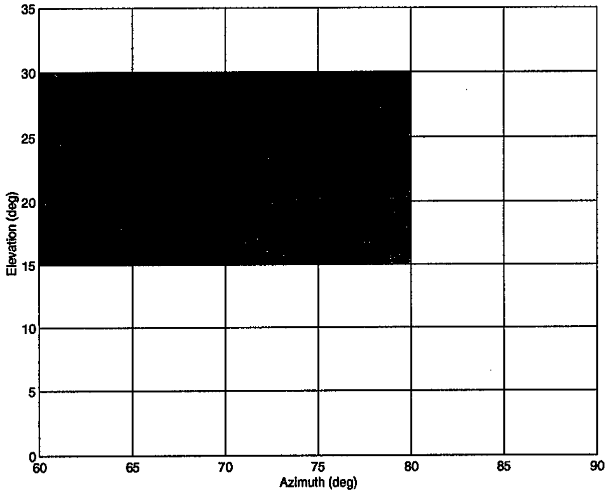


Figure 4.22 Span of HRR data used for testing.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	2048	255	66	8	0	0	86.2
B	0	1563	0	0	0	0	100.0
C	42	3	522	352	0	87	51.9
D	0	89	1	1333	0	0	93.7
E	68	55	28	376	64	141	8.7
F	7	81	0	187	11	773	73.0
All Classes:							68.9

Table 4.41 Cummulative baseline target confusion matrix for the case of 12 windows, ten looks, and synthetic training data.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	1755	378	195	13	1	35	73.8
B	15	1536	0	8	0	4	98.3
C	18	7	738	123	44	76	73.4
D	22	183	97	990	26	105	69.6
E	34	47	150	96	323	82	44.1
F	26	51	92	76	60	754	71.2
All Classes:							71.7

Table 4.42 Cummulative target confusion matrix with denoising for the case of 12 windows, one look, and synthetic training data.

Actual Class	Assigned Class						P_C (%)
	A	B	C	D	E	F	
A	1985	206	175	0	0	11	83.5
B	0	1563	0	0	0	0	100.0
C	0	0	947	14	6	39	94.1
D	0	100	57	1254	3	9	88.1
E	1	40	166	63	409	53	55.9
F	2	12	13	38	73	921	87.0
All Classes:							84.8

Table 4.43 Cummulative target confusion matrix with denoising for the case of 12 windows, ten looks, and synthetic training data.

	Targets						
# Looks	A	B	C	D	E	F	Avg
1 Look	-1.4	0.4	29.6	-8.8	31.3	10.2	10.2
10 Looks	-2.7	0.0	42.2	-5.6	47.1	14.0	15.9

Table 4.44 Relative classification improvements for the case of 12 windows, synthetic training data, and all available testing data.

each decomposition level, all wavelets that are contained in the parameter space, with the remaining parameters fixed to those in Table 4.10 and obtain classification accuracies for a single look. The assessment is not intended to be rigorous by any means and is primarily for qualitative purposes. In examining sensitivity for this particular case of synthetic training data and one look, we assume that the results generalize for other testing cases. See Figure 4.23

We see that the choice of wavelet has a substantial effect on classification performance and that we are justified in optimizing over it. Notice in Figure 4.23 that at coarser wavelet scales smoother wavelets become optimal, where smoothness is defined in terms of the K-Regular Scaling Filters that were briefly mentioned in Chapter 2. We attribute no significance to this however, since we did not see the same trend in the case of multiple windows.

4.4.3 Alternative Denoising Methods. In Chapter 2 we discussed wavelet based denoising methods and in Chapter 3 we enumerated reasons for abandoning those methods which lead us to the methodology which has formed the basis of this thesis. Recall that

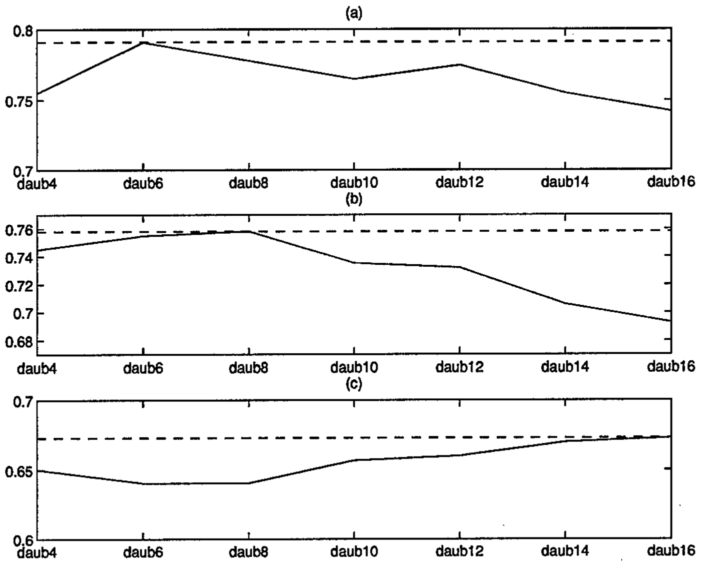


Figure 4.23 Illustration of the sensitivity of classification accuracy with respect to the wavelet choice for the case of a single window, single look, and synthetic training data.

traditional wavelet denoising is based on the Gaussian noise model and that we adopt an abstract notion of noise which forces us to pursue optimal denoising with respect to classification accuracy as opposed to a risk measure (such as MSE). We are interested in determining what we gain by viewing noise in this manner, as opposed to the Gaussian noise model in which case the VisuShrink and SureShrink methods are preferable.

The VisuShrink and SureShrink denoising methods assume a signal model $s_i = f_i + \sigma z_i$, $i = 1, 2, \dots, N$, where f is a deterministic signal and $\{z_i\}$ are distributed as $z \overset{iid}{\sim} N(0, 1)$. The risk measure is MSE between f and $\hat{f}$. Both are performed in three steps:

1. Compute wavelet transform.
2. Apply soft thresholding to detail coefficients.
3. Reconstruct to obtain $\hat{f}$.

In step 2, the threshold is chosen as described in Chapter 2. Denoising is implemented using the third-party Matlab toolbox WaveLab, as was the case with the implementation of the TI transform. As with the TI denoising method of this thesis, we allow the choice of wavelet to vary and we perform denoising at decomposition levels one through seven.

However, we allow the decomposition to progress to coarser scales than was the case with the TI method. The VisuShrink and SureShrink methods are optimized over the wavelet choice using a single window and a single look. Figure 4.24 shows the results.

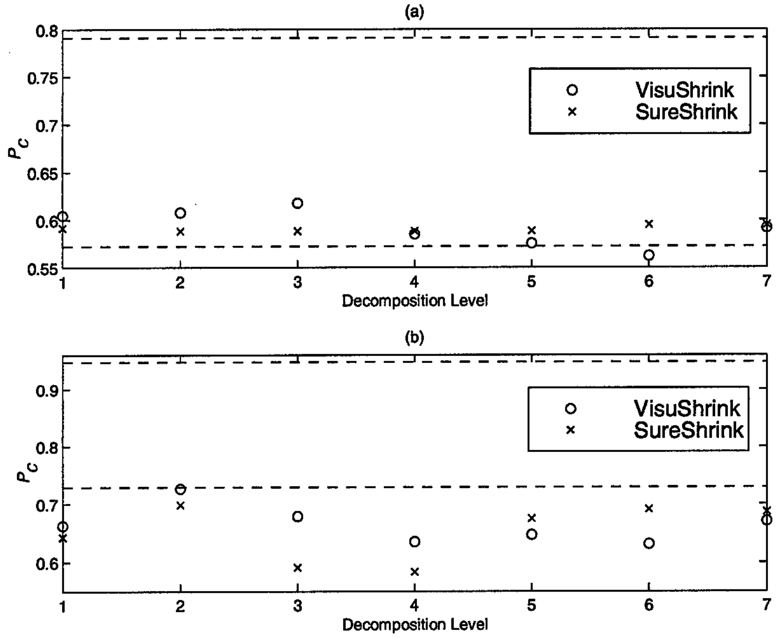


Figure 4.24 Maximum VisuShrink and SureShrink overall target accuracy as a function of decomposition level for the case of a single window and synthetic training data. Top line corresponds to maximum accuracy achieved through TI denoising; bottom line corresponds to baseline accuracy. (a) Single look; (b) Ten looks

We are most interested in the ten look results and we see that at best, VisuShrink and SureShrink are able to match the baseline performance. Let us now examine representative signatures obtained through VisuShrink and SureShrink denoising. See Figures 4.25 and 4.26.

We see that both denoising methods do not significantly alter the original HRR signature. VisuShrink exhibits a larger degree of smoothing than SureShrink as is characteristic of VisuShrink. Still, it is not a large enough extent. We conclude that the TI denoising method provides us with superior results due to the large degree of smoothing that it affords. This capability exists due to the thresholding of approximation coefficients.

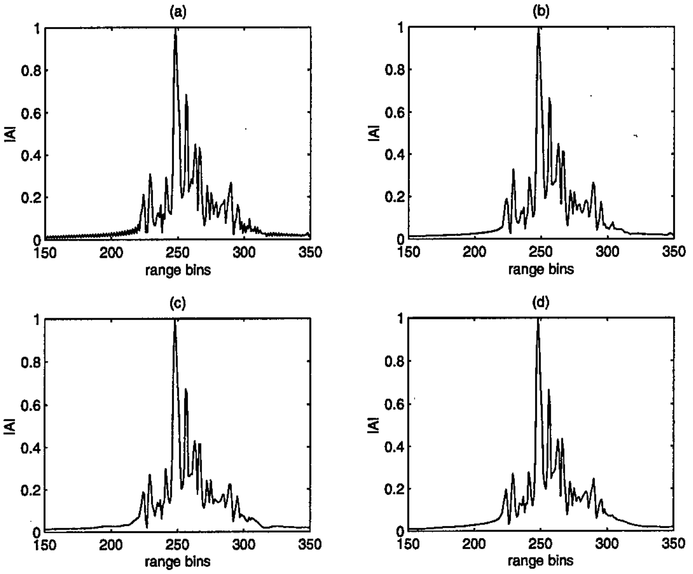


Figure 4.25 Denoised signal representations obtained through VisuShrink for the case of a single window and synthetic training data. (a) Original HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signature using level 4 parameters; (d) Denoised signature using level 7 parameters;

Let us now consider denoising using finite impulse response (FIR) filtering. A FIR filter has a transfer function of the form

$$H(z) = \sum_{k=0}^{N-1} h(k) z^{-k}, \tag{4.4}$$

where N is the filter length. The filtering (denoising) of a HRR signature $s(k)$ can be implemented as a standard difference equation as

$$f(k) = \sum_{i=0}^{N-1} h(k) s(k-i), \tag{4.5}$$

where f is the filtered (denoised) signal. The details of designing such a filter are in (42). We optimize overall target accuracy for the case of a single window, a single look, and synthetic training data. The denoising parameters are the filter length and the cutoff frequency which is in normalized frequency. We can view an accuracy surface as a function of the filter parameters as in Figure 4.27. The lower plane indicates baseline accuracy and the upper plane indicates the highest accuracy that was achieved with the TI denoising

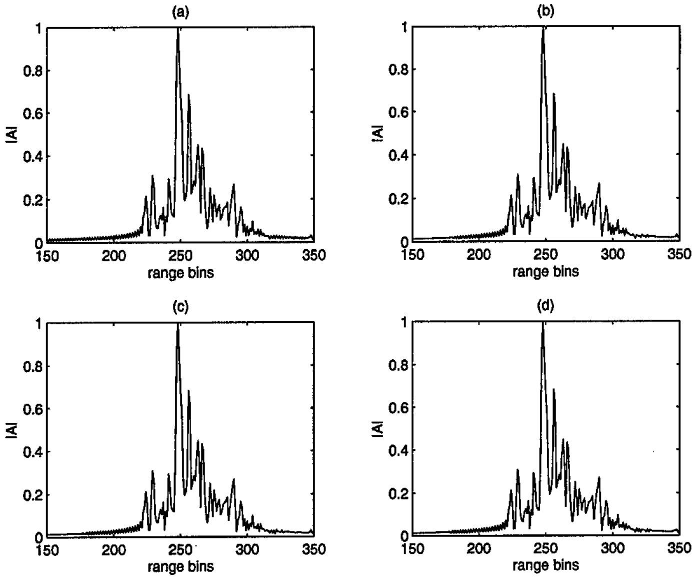


Figure 4.26 Denoised signal representations obtained through SureShrink for the case of a single window and synthetic training data. (a) Original HRR signature; (b) Denoised signature using level 1 parameters; (c) Denoised signature using level 4 parameters; (d) Denoised signature using level 7 parameters;

scheme. We see that FIR filtering does not lead to a significant increase in classifier performance. A maximum accuracy of approximately 61.2% is obtained with a filter length of 9 and a normalized cutoff frequency of 0.7. Figure 4.28 shows a typical signal representation obtained through FIR filtering with the optimal parameters. Compare this representation with those obtained through VisuShrink and SureShrink as seen in Figures 4.25 and 4.26. Note that the traditional wavelet methods and the FIR method both result in representations that are nearly identical, visually. This is not surprising since they both achieve similar accuracies in the single look classification case. This observation leads us to believe that FIR filtering does not lead to improved results for the case of multiple looks and so multiple look performance is not investigated. This is a valid belief since VisuShrink and SureShrink were able to match ten look baseline performance at best.

4.4.4 Variation of the Denoising Implementation. Are we really gaining anything by performing the translation invariant wavelet transform as opposed to the standard wavelet transform? We can easily examine this issue by substituting the standard wavelet transform for the TI wavelet transform, and optimizing over the established parameter

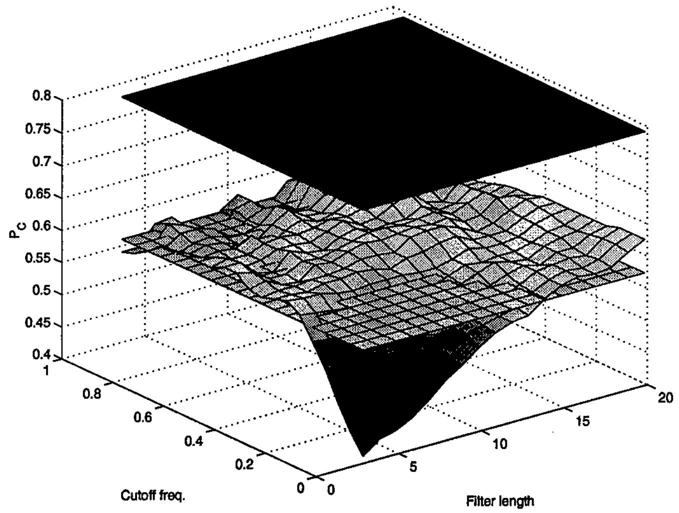


Figure 4.27 Visualization of overall classification accuracies obtained with FIR filtering for the case of a single window, single look, and synthetic training data.

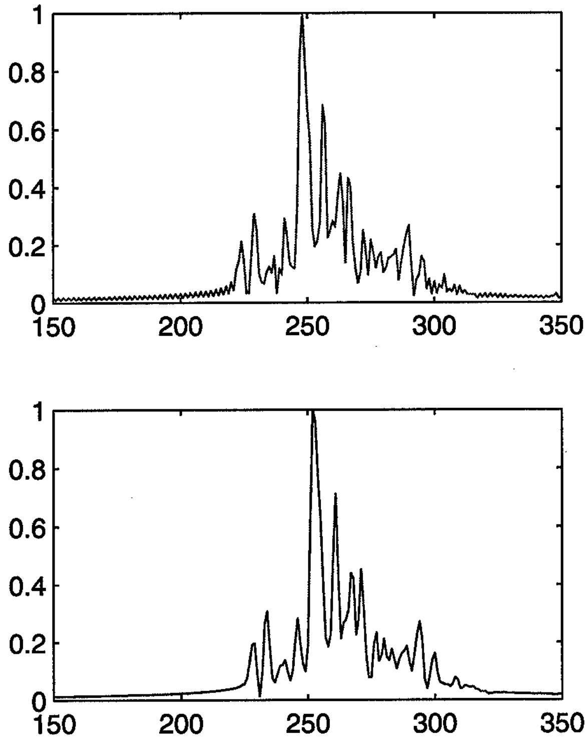


Figure 4.28 Original HRR signature (top) and denoised signature (bottom) obtained with optimal FIR filtering.

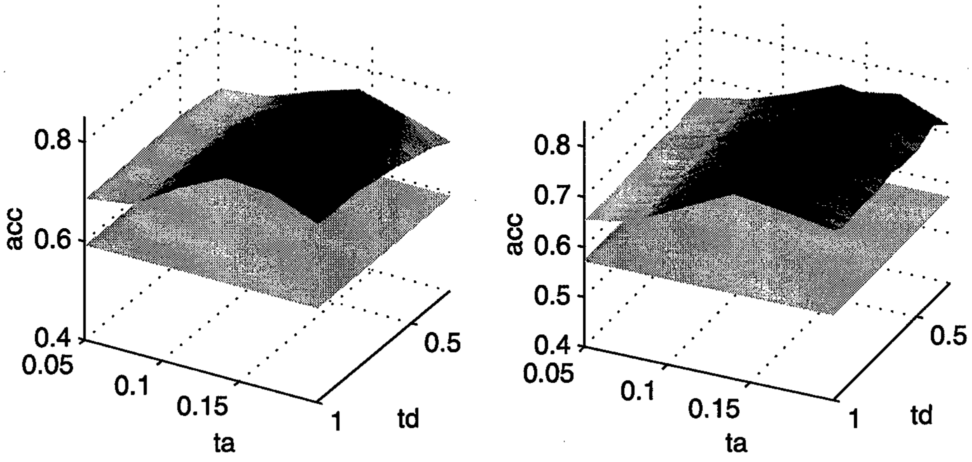


Figure 4.29 Level one denoising accuracy surfaces for the case of a single window and synthetic training data. Left: TI method; Right: Non-TI method

space. Conceptually, this is equivalent to retaining the topmost coefficient collections in each column of the TI Table, and setting all others to zero. That is, we retain the coefficients $\mathbf{a}_{j_0,1}, \mathbf{d}_{J-1,1}, \mathbf{d}_{J-2,1}, \ldots, \mathbf{d}_{j_0,1}$ which comprise the standard wavelet transform. For illustration purposes, we do this for only one decomposition level. Let us view the resulting accuracy surface along with that of the TI method. These are shown in Figure 4.29. We see that both methods are nearly equivalent. The non-TI method achieves a maximum accuracy of 77.8% with $daub_{16}$, soft thresholding, $t_a = 0.135$, and $t_d = 0.85$. Note that these thresholds are nearly identical to those obtained for the TI method. The non-TI implementation accuracy is almost 2% worse than the accuracy obtained with the TI method. Though this difference may not be statistically significant, we still prefer the TI method because of the desirable property that $\hat{f}_s = S_s\hat{f}$, as mentioned in Chapter 2. The results in this section also support the claim that the approximation coefficient thresholding is what provides the significant performance improvement.

4.4.5 Alternative Optimization Procedure for Multiple Windows. Recall that when we optimized the denoising method over multiple windows, that the aim was to maximize the average overall accuracy across all windows using a single set of denoising parameters. One may wonder why we did not optimize with respect to windows individually such that separate parameters would be determined for each window. We chose not

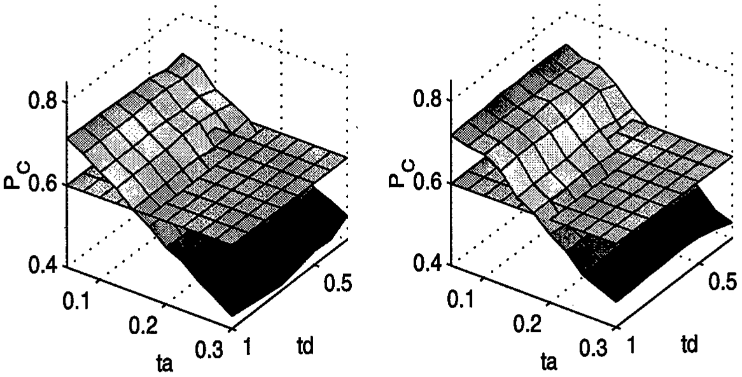


Figure 4.30 Visualization of maximum classification accuracies as a function of threshold pairs for the case of $win_{60,15}$ and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2

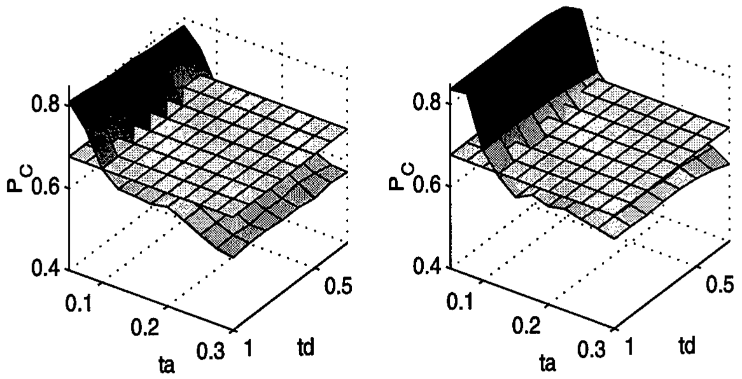


Figure 4.31 Visualization of maximum classification accuracies as a function of threshold pairs for the case of $win_{60,25}$ and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2

to do so primarily for simplicity. Using a single set of optimal parameters to achieve improvement across all windows is preferred over using separate parameters for each window because it is likely to be more robust. We now revisit the case of multiple windows and synthetic training data and optimize each window separately. We only need to optimize for four of the windows since optimization was already done for $win_{65,15}$. Let us begin in the usual manner by viewing accuracy surfaces. As a simplification, we only consider decomposition levels one and two.

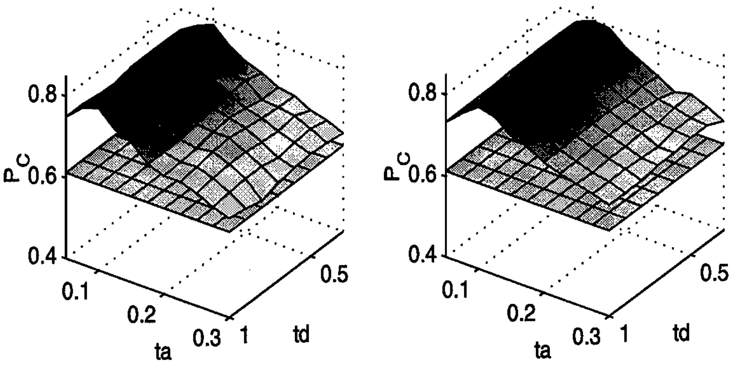


Figure 4.32 Visualization of maximum classification accuracies as a function of threshold pairs for the case of $win_{70,15}$ and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2

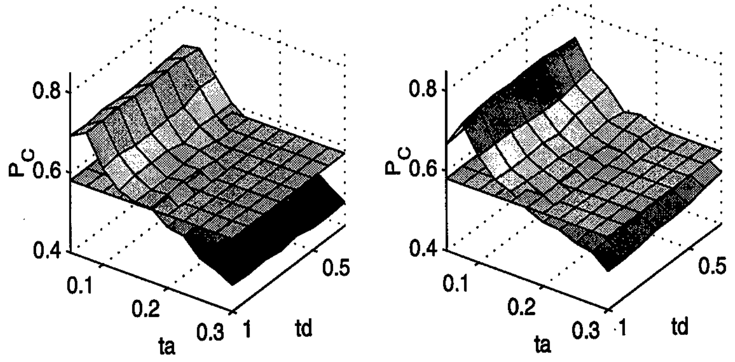


Figure 4.33 Visualization of maximum classification accuracies as a function of threshold pairs for the case of $win_{75,15}$ and synthetic training data. (a) Decomposition level 1; (b) Decomposition level 2

Window	**Wavelet**	**Thresholding Method**	t_a	t_d
$win_{60,15}$	$daub_4$	soft	0.050	0.70
$win_{60,25}$	$daub_6$	soft	0.050	1.00
$win_{65,15}$	$daub_6$	soft	0.195	0.70
$win_{70,15}$	$daub_{12}$	soft	0.100	1.00
$win_{75,15}$	$daub_{10}$	soft	0.075	0.80

Table 4.45 Optimal level two denoising parameters for the case of five windows and synthetic training data

	Accuracies (P_C)		
Window	**Collectively**	**Separately**	**Improvement**
$win_{60,15}$	71.9	72.5	0.6
$win_{60,25}$	85.4	85.5	0.1
$win_{65,15}$	69.6	75.8	6.2
$win_{70,15}$	77.3	82.7	5.4
$win_{75,15}$	72.6	75.9	3.3
Avg.	75.8	78.7	2.9

Table 4.46 Comparison of overall target accuracies for different multiple window optimization techniques

From these surfaces we see that accuracies either decrease nearly monotonically with increasing t_a, or the accuracies rise then fall with increasing t_a. This observation suggests that 5 X 5 windows can be broken into two major categories. It is possible that we could determine two sets of denoising parameters, each achieving near optimal performance for the respective window group.

We find that maximum accuracies occur at the second decomposition level for all windows (with the exception of $win_{65,15}$). Optimal parameters are shown in Table 4.45. Now let us compare the overall target accuracies obtained with both optimization methods. See Table 4.46. Note that optimizing windows individually provides us with a gain of about only 3%. This suggests that optimizing over all windows simultaneously may be the preferable method since it possesses a strong degree of robustness.

4.5 Summary

In this chapter we have seen some remarkable results. We demonstrated that we can achieve HRR classification accuracies equivalent to those of the baseline classifier when training on measured data, through a denoising scheme that affords us simpler signal representations. More importantly though, we have achieved enormous accuracy improvements when training on synthetic data. When incorporating multiple looks, the denoising scheme leads to classification accuracies which approach those of the measured training data case. These results have far reaching implications and we now conclude this research and make recommendations.

V. Conclusions and Recommendations

5.1 Introduction

The primary goal of this thesis is to achieve substantial improvement in HRR classification for synthetic training data. Improvements in the synthetic case often lead to degradation in the measured case, and thus a secondary goal is to maintain the baseline performance when training on measured data. We first approach this challenging problem through visualization of the raw HRR signatures. The visual observations motivate us to apply a sound philosophy: we prefer simpler models to complex ones. We view the signals as containing noise, but the concept of noise extends beyond the standard Gaussian noise model. Removal of Gaussian noise would simplify the HRR signatures, but in the framework of this thesis, the assertion is that such removal would not be extensive enough to provide us with the simple signal representations that we seek. An abstract noise model is then adopted and we decide that, in general, any quality of the signals that prevents classification improvement is noise. We develope a powerful wavelet-based denoising scheme that allows us to consider a larger class of signal representations than would be possible with standard wavelet-based techniques. We implement a computationally efficient wavelet transform that has desirable translation invariance properties. Abandonment of the Gaussian noise model forces us to optimize the denoising parameters with respect to classification accuracy and this optimization is accomplished through an exhaustive search of the parameter space. Ultimately, the denoising scheme enables us to achieve *remarkable* improvements in classification accuracy when using synthetic training data. We find that the denoised signals are indeed much simpler than the original signals and that the abstract notion of noise is a powerful viewpoint.

5.2 Summary of Key Results

We summarize key results of this thesis:

- A powerful wavelet-based denoising method is developed and serves as a pre-processing step in HRR classification. This method is optimized with regard to classification accuracy in the context of an abstract noise model.

- The denoising method enables us to achieve excellent classification results equivalent to those of the baseline classifier for a single azimuth and elevation window and measured training data. We find that equivalent performance is achieved with simpler signal representations. When training on synthetic training data, we obtain *remarkable* classification improvements which match those obtained for measured training data. This result is unprecedented.

- Generalization and robustness of the denoising performance is demonstrated by first optimizing and testing over multiple windows. Results are, in general, similar to those found for the single window case. As a more rigorous means of assessing generalization and robustness, we classify *all* available testing data from 12 windows using the denoising parameters determined from optimization over five windows. We do this for the case of synthetic training data only since this is the most relevant case. Classification under these circumstances yields performance gains similar to those found for the five window case. The rigorous testing demonstrates, with confidence, that the denoising methodology of this thesis is an *unquestionable* means to achieve significant classification improvement when using synthetic data.

- The denoising method of this thesis is shown to achieve classification results superior to those obtainable through traditional wavelet-based methods. A key factor enabling superiority is the decision to consider noise abstractly, as opposed to the traditional approach of specifying a Gaussian noise model. The abstract view of noise provides justification in thresholding approximation coefficients, since we do not necessarily want to maintain the underlying signal structure. The denoising methodology determines the required underlying structure by virtue of the optimization process.

5.3 Recommendations for Future Work

The results of this thesis warrant considerable further research:

- Modify the denoising scheme so that individual detail thresholds are specified for each decomposition level. The interest here is in performance gains (if any) and in whether or not the additional parameters cause a loss in robustness.

- Perform extensive testing of the denoising scheme across all 5 X 5 windows for which there is a sufficient number of training data. The testing can be done in one of three ways, all which should be examined: 1) Optimize the denoising method for windows individually. When testing an unknown signature, the denoising parameters associated with the corresponding 5 X 5 window are used. 2) Optimize across all windows simultaneously so that maximum average overall accuracy is obtained. 3) Characterize the accuracy surfaces found in 1). It is likely that a small number of window groups will exhibit similar accuracy surfaces based on results in the previous chapter. Each group can then be assigned a unique set of optimal denoising parameters. If there is a window for which there is little data, then assign it to the group for which the accuracy surfaces are most regularized. This scheme has great potential in that it is likely to result in a higher average accuracy than in 2). It is also more concise than 1) and is likely to be more robust.

- Modify denoising methodology so that parameters can be determined individually for targets. This approach could alleviate the problem seen in Chapter 4, in which case target E signatures contained residual, uninformative peaks following denoising. Though these peaks do not seem to affect classification performance, it is still desirable to remove them since doing so results in scatter plots which are more amiable to post-processing. This approach also has the potential to eliminate the degradation in certain targets caused by denoising, as seen in Chapter 4.

- Design wavelet systems whose basis functions more closely resemble the HRR signatures than do standard wavelet basis functions. This design might be approached from an eigenvalue/eigenvector standpoint.

- Use the denoising methodology of this thesis as a preliminary step in a subsequent piecewise polynomial fitting routine. Investigate the use of the piecewise parameters as features for classification. Since this thesis (and others preceding it) demonstrates that large amounts of signature information can be discarded, it seems reasonable that we can represent signatures in a simple, piecewise manner.

5.4 Summary

This thesis makes a major contribution to the HRR classification problem. Instead of trying to modify (or replace) the baseline classifier, we approach the problem solely from a pre-processing standpoint as a means to merely augment the baseline classifier. The results suggest that further work in this area focus on the problem in the context of pre-processing. The unprecedented accuracies obtained when training on synthetic data suggest that synthetic and measured data can be successfully integrated into a classification system, thereby satisfying one of the primary goals of the NCTI community. The results in this thesis also raise some philosophical issues. Since simpler signal representations arise from denoising, we may consider HRR classification as an inverse problem: How do we build radar systems that acquire signatures with forms similar to those seen in this thesis? Does high range resolution really provide benefits, given that we a) achieve equivalent performance when training on measured data by using much simpler representations, and b) achieve remarkable performance improvement when training on synthetic data with simpler signal representations? Initial high fidelity signals may be needed for reduction to the forms observed in this thesis. If this is the case, then the wavelet pre-processing of this thesis is a very attractive means to obtain simplified signal representations. We have used the word "simple" and its various forms throughout this thesis because we desire to convey the advantages gained from reducing the complexity of problems. Perhaps in simplifying problems we are regressing, but the underlying results of this thesis suggest that it is such a regression that is needed to satisfy the aims of HRR classification.

Bibliography

1. Abramovich, Felix and Yoav Benjamin. "Thresholding of Wavelet Coefficients as Multiple Hypotheses Testing Procedure." *Wavelets and Statistics* edited by Anestis Antoniadis and Georges Oppenheim, 5–14, Springer-Verlag, 1995.
2. Akansu, Ali N. and Richard A. Haddad. *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*. Academic Press, 1992.
3. Baras, John and Sheldon Wolk. "Wavelet Based Progressive Classification with Learning: Applications to Radar Signals," *SPIE*, *2491*:339–349 (1995).
4. Bishop, Christopher M. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
5. Broussard, Randy, January 1999. Conversation.
6. Bruce, Andrew and others. "Denoising and Robust Non-Linear Wavelet Analysis," *SPIE*, *2242*:325–336 (1994).
7. Buckheit, J. B. and D. L. Donoho. "Wavelab and Reproducible Research." *Wavelets and Statistics* edited by Anestis Antoniadis and Georges Oppenheim, 55–81, Springer-Verlag, 1995.
8. Burrus, C. Sidney and others. *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall, 1998.
9. Carmona, Rene. "Wavelet Identification of Transients in Noisy Time Series," *SPIE*, *2034*:392–400 (1993).
10. Carmona, Rene and L. Hudgins. "Wavelet Denoising of EEG Signals and Identification of Evoked Response Potentials," *SPIE*, *2303*:91–104 (1994).
11. Cena, Bernard and Nick Spadaccini. *Wavelet Shrinkage of Ultrasound Data*. Technical Report, Department of Computer Science, The University of Western Australia, 1996.
12. Chan, Y.T. *Wavelet Basics*. Kluwer Academic Publishers, 1995.
13. Cohen, Marvin. "Variability of Ultra-High Resolution Radar Profiles and Some Implications for Target Recognition," *SPIE*, *1699*:256–266 (1992).
14. Coifman, Ronald R. and Mladen Victor Wickerhauser. "Entropy-Based Algorithms for Best Basis Selection," *IEEE Transactions on Information Theory*, *38*(2):713–718 (March 1992).
15. Coifman, R.R. and D.L. Donoho. "Translation-Invariant De-Noising." *Wavelets and Statistics* edited by Anestis Antoniadis and Georges Oppenheim, 125–150, Springer-Verlag, 1995.
16. Cranos, Roger, "Combat Identification in the Future: Maintaining a Balance." Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) 11th Annual International Symposium on Aerospace/Defense Sensing.

17. Daqing, Chen and Bao Zheng. "High Range Resolution Radar Target Identification Using Multilayer Feedforward Neural Network." *CIE International Conference of Radar Proceedings*. 215–218. 1996.

18. Deng, Lian Ping. "Radar Target Recogntion with Fractal Technology," *SPIE*, *2759*:511–517 (1996).

19. Dewall, Robert, February 1999. Conversation.

20. Donoho, David and Iain Johnstone. "Ideal Spatial Adaptation by Wavelet Shrinkage," *Biometrika*, *81*:425–455 (1994).

21. Donoho, David and Iain Johnstone. "Adapting to Unknown Smoothness via Wavelet Shrinkage," *Journal of the American Statistical Association*, *90*:1200–1224 (1995).

22. Donoho, David L. "Unconditional Bases are Optimal Bases for Data Compression and for Statistical Estimation," *Applied and Computational Harmonic Analysis*, *1*(1):100–115 (December 1992).

23. Donoho, David L. "De-Noising by Soft-Thresholding," *IEEE Transactions on Information Theory*, *41*(3):613–627 (May 1995).

24. Eisenbies, Christopher L. *Classification of Ultra High Range Resolution Radar Using Decision Boundary Analysis*. MS thesis, Air Force Institute of Technology, 1994.

25. Eom, Kie and Rama Chellapa. "Noncooperative Target Classification Using Hierarchial Modeling of High Range Resolution Radar Signatures," *IEEE Transactions on Signal Processing*, *45*(9):2318–2327 (1997).

26. Fukunaga, Keinosuke. *Introduction to Statistical Pattern Recognition* (Second Edition). Academic Press, 1990.

27. Ghael, Sandeep and others. "Improved Wavelet Denoising via Empirical Wiener Filtering," *SPIE*, *3169*:389–399 (1997).

28. He, Songhua and others. "High Range Resolution MMW Radar Target Recognition Approaches with Application." *IEEE Proceedings of the National Aerospace and Electronics Conference1*. 192–195. 1996.

29. Hu, Rong and Zhaoda Zhu. "Researches on Radar Target Classification Based on High Range Resolution Range Profiles." *Proceedings of the 1997 IEEE National Aerospace and Electronics Conference2*. 951–955. 1997.

30. Huaitie, Xiao and others. "Aircraft Target Recognition Using Adaptive Time Delay Neural Network." *Proceedings of the 1997 IEEE National Aerospace and Electronics Conference2*. 764–768. 1997.

31. Kay, Steven M. *Fundamanetals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

32. Kosir, Peter, January 1999. Conversation.

33. MacDonald, Adam. *Classification of High Range Resolution Radar Returns Using Hidden Markov Models and Gaussian Mixture Models*. MS thesis, Air Force Institute of Technology, 1997. Unpublished.

34. Mallat, Stephanie. "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*:674–693 (1989).

35. Nason, G. P. *Wavelet Regression by Cross Validation*. Technical Report, Department of Mathematics, University of Bristol, March 1994.

36. Nason, G. P. "Choice of the Threshold Parameter in Wavelet Function Estimation." *Wavelets and Statistics* edited by Anestis Antoniadis and Georges Oppenheim, 261–280, Springer-Verlag, 1995.

37. Nason, G. P. and B. W. Silverman. "The Stationary Wavelet Transform and Some Statistical Applications." *Wavelets and Statistics* edited by Anestis Antoniadis and Georges Oppenheim, 281–299, Springer-Verlag, 1995.

38. Nason, G. P. and B. W. Silverman. *Wavelets for Regression and Other Statistical Problems*. Technical Report, Department of Mathematics, University of Bristol, June 1997.

39. Ogden, R. Todd. *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhauser, 1997.

40. Pham, Dzung Tri. *Applications of Unsupervised Clustering Algorithms to Aircraft Identification Using High Range Resolution Radar*. MS thesis, Air Force Institute of Technology, 1997.

41. Pin, Chen Zhen and others. "Radar Target Identification of Aircraft Using Structural Features by Correlation Method." *Proceedings of the IEEE 1995 National Aerospace and Electronics Conference 2*. 114–116. 1995.

42. Roberts, Richard and Clifford Mullis. *Digital Signal Processing*. Addison Wesley, 1987.

43. Stewart, Clayton and Others. "Comparison of Classification Approaches for High Range Resolution Radar," *SPIE*, *1700*:146–155 (1992).

44. Therrien, Charles W. *Decision, Estimation and Classification*. John Wiley and Sons, 1989.

45. Ulug, Batuhan and others. "Model-Catalog Compression for Radar Target Recognition." *IEEE International Conference on Acoustics, Speech and Signal Processing 5*. 3479–3482. 1995.

46. Wehner, Donald. *High Resolution Radar*. Artech House, 1987.

47. Weiss, L. G. and T. L. Dixon. "Wavelet-Based Denoising of Underwater Acoustic Signals," *Journal of the Acoustical Society of America*, *101*(1):377–382 (1997).

48. Wilbur B. Davenport, Jr. *Probability and Random Processes: An Introduction for Applied Scientists and Engineers*. McGraw Hill, 1970.

49. Xun, Zhang and others. "Radar Target Recognition Using Wavelet Transform and Information Fusion." *International Conference on Signal Processing Proceedings 1*. 355–358. 1996.

50. Xun, Zhang and others. "Automatic HRR Target Recognition Based on Matrix Pencil Method and Multiresolution Neural Network," *SPIE*, *3069*:510–517 (1997).

51. Xun, Zhang and others. "HRR Target Recognition Using the Geometry Information of Scattering Centers." *Proceedings of the 1997 IEEE National Aerospace and Electronics Conference2*. 936–940. 1997.

52. Yicheng, Jiang and others. "A New Radar Target Classification Approach Based on Polarimetric High Range Resolution." *CIE International Conference of Radar Proceedings*. 147–150. 1996.

53. Zhao, Qun and others. "Radar Target Identification Using Wavelet Transform of Range Profiles." *Proceedings of the International Conference on Neural Information Processing1*. 423–427. 1996.

54. Zumwalt, Michael P. *Robust High Range Resolution Radar Classification Using Synthetic Data With Multinomial Pattern Matching*. MS thesis, Air Force Institute of Technology, 1998. Unpublished.

55. Zywaeck, Anthony and Robert Bogner. "Radar Target Recognition Using Range Profiles." *IEEE International Conference on Acoustics, Speech and Signal Processing2*. 373–376. 1994.

Vita

Brian Michael Huether was born on 11 October 1973 in Lowell, Massachusettes. He received a Bachelor of Science in Electrical Engineering at the University of New Hampshire, and graduated *Magna Cum Laude* in May 1997. While attending college, he was enrolled in the Air Force Reserve Officers Training Corps (ROTC) program, through which he recieved a commission as a Second Lieutenant upon graduation. Lieutenant Huether was awarded a direct accession into the Air Force Institute of Technology (AFIT), Wright Patterson Air Force Base, Ohio, as his first assignment. In March 1999, Lieutenant Huether graduated from AFIT, having received a Master of Science in Electrical Engineering. In April 1999, Lieutenant Huether will report to the National Aerospace Intelligence Center (NAIC) at Wright Patterson, where he will serve as a Ground Analyst in the Technology Assessment, Space Systems Division.

Permanent address: 169 Jeremy Hill Rd.
Pelham, NH 03076